FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

IPBrick - Módulo de diagnóstico e de despiste de problemas

Luís Carlos Ferreira Borges

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: João Manuel Couto das Neves

Co-orientador: Miguel Ramalhão Ribeiro

30 de Junho de 2014

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES | **2013/2014**

A Dissertação intitulada

"IPBrick - Módulo de Diagnóstico e de Despiste de Problemas"

foi aprovada em provas realizadas em 16-07-2014

o júri

Presidente Professor Doutor Paulo José Lopes Machado Portugal
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Carlos Leite Ramalho
Professor Auxiliar do Departamento de Informática da Escola de Engenharia da Universidade do Minho

Professor Doutor João Manuel Couto das Neves
Professor Auxiliar Convidado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Luís Carlos Ferreira Borges

Faculdade de Engenharia da Universidade do Porto

Resumo

A Internet é composta pela união de diversas redes a nível mundial. A ligação entre essas redes e entre todos os dispositivos ligados às mesmas, permite que um elevado número de utilizadores tenha acesso a recursos disponíveis em servidores. e.g. páginas *web* e ficheiros. De facto, os servidores assumem uma posição central e fundamental nesta estrutura da Internet.

Uma falha em qualquer servidor provoca elevados transtornos para os utilizadores. As falhas não detetadas podem levar a comportamentos imprevisíveis do sistema, ou mesmo à danificação do *hardware* do servidor. Só a deteção atempada permite uma rápida resolução. Esta razão, aliada à crescente importância destes servidores, torna crucial a minimização do tempo de inatividade.

A IPBrick é uma solução de servidores desenvolvido pela empresa IPBrick SA, que se caracteriza por apresentar um sistema operativo baseado no Linux Debian e oferecer uma interface *web* multifuncional através da qual o utilizador pode configurar, de um modo simples e intuitivo, todo o sistema.

Devido ao facto da solução IPBrick não incluir nenhuma ferramenta de diagnóstico que permita identificar a origem das falhas, surgiu a necessidade de conceber um módulo capaz de realizar um *check-up* ao sistema que identifique a origem dos problemas de forma a agilizar a sua resolução.

A realização desta dissertação, visa o desenvolvimento de um módulo que permite efectuar a deteção de falhas na solução IPBrick. O desenvolvimento do módulo está inserido num projeto proposto pela empresa IPBrick SA à Faculdade de Engenharia da Universidade do Porto.

Abstract

The Internet is composed by the union of several networks worldwide. The link between these networks and between all devices connected to them, allows a large number of users to have access to resources available on servers. E.g. web pages and files. Indeed, the servers assume a central and pivotal position in this structure of the Internet.

A failure in any server causes high inconvenience to users. The undetected failures can lead to unpredictable behavior of the system, or even damage the server hardware. Only the timely detection allows a quick resolution. This reason, coupled with the increasing importance of these servers becomes crucial to minimize downtime.

IPBrick is a server solution developed by the company IPBrick SA, which is characterized by having a system based on the Debian Linux operating system and offer a multifunctional web interface through which the user can set, in a simple and intuitive way, the entire system.

Because the solution IPBrick does not include any diagnostic tool for identifying the origin of failures, it became necessary to think in a module able to perform a complete check-up to the system to identify the source of problems in order to expedite its resolution.

The main goal of this dissertation is to develop a module that execute the failure detection in IPBrick solution. The development of the module is inserted into a project proposed by the company IPBrick SA to the Faculty of Engineering of University of Porto.

Agradecimentos

Primeiramente, gostaria de agradecer aos orientadores que me acompanharam ao longo do desenvolvimento deste projeto, Professor Doutor João Neves e Engenheiro Miguel Ramalhão.

Gostaria, também, de agradecer aos meus pais pelo apoio prestado e pelo esforço que realizaram para a concretização de mais uma etapa na minha vida.

Por último, agradeço a todas as pessoas que me apoiaram e que, de certa forma, contribuíram para a realização deste trabalho.

Luís Borges

"People might not get all they work for in this world,
but they must certainly work for all they get."

Fredrick Douglass

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos e Abreviaturas

ACK	Acknowledgement
CPU	Central processing unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
GNU GPL	GNU General Public License
HTTP	Hypertext Transfer Protocol
ICANN	Internet Corporation for Assigned Names and Numbers
IM	Intant Messaging
IMAP	Internet Message Access Protocol
IMAPs	Internet Message Access Protocol Secure
IP	Internet Protocol
IPPBX	Internet Protocol Private Branch Exchange
ISP	Internet service provider
LAN	Local area network
LDAP	Lightweight Directory Access Protocol
MDA	Mail Delivery Agent
MIB	Management Information Base
MS-DOS	MicroSoft Disk Operating System
MTA	Mail Transfer Agent
MUA	Mail User Agent
NRPE	Nagios Remote Plugin Executor
OID	Object ID
PBX	Private Branch Exchange
PHP	Hypertext PreProcessor
POP	Post Office Protocol
POP3	Post Office Protocol v3
POPs	Post Office Protocol Secure
PSTN	Public Switched Telephone Network
RAM	Random Access Memory
SIP	Session Initiation Protocol
SIPs	Session Initiation Protocol Secure
SMTP	Simple Mail Transfer Protocol
SMTPs	Simple Mail Transfer Protocol Secure
SNMP	Simple Network Management Protocol
SQL	Structured Query Languag
SSL	Secure Sockets Layer
UCoIP	Unified Communications over IP

URL	Uniform Resource Locator
VoIP	Voice over IP
VPN	Virtual Private Network
WWW	World Wide Web

Capítulo 1

Introdução

Neste capitulo apresenta-se a motivação para este trabalho, acompanhada de uma breve descrição do tema. Por fim, clarifica-se a estrutura do presente documento.

1.1 Motivação

A *Internet*, como hoje é conhecida, é vista como uma interligação de diversas redes a nível mundial. A ligação entre essas redes e, por conseguinte, entre todos os dispositivos ligados às mesmas, permite que um elevado número de utilizadores tenha acesso a recursos disponíveis em servidores. e.g. páginas *web* e ficheiros.

As falhas em qualquer servidor ocorrem quando, por algum motivo, este é impedido de executar as tarefas para que foi destinado. As falhas não detetadas podem levar a comportamentos imprevisíveis do sistema, ou mesmo à danificação do *hardware* do servidor. Só a deteção a tempo das falhas permite agilizar a resolução. Esta razão, aliada à crescente importância destes servidores, torna crucial a minimização do tempo de inatividade.

A presente dissertação prevê a especificação de um módulo IPBrick que permite detetar as anomalias impeditivas do funcionamento correto dos servidores, de modo a acelerar a sua resolução.

1.2 Tema

O sistema operativo IPBrick consiste numa solução completa de servidores cujo principal público alvo são as organizações empresariais. A característica que a distingue de outras soluções é a facilidade concedida ao utilizador no ato de configuração. Contribui para esta facilidade, a disponibilização, ao utilizador, de uma interface *web* simples e intuitiva que permite configurar todos os parâmetros do servidor facilmente.

Um dos pontos fracos da solução IPBrick prende-se com a incapacidade de deteção de falhas de uma forma rápida e eficaz. Atualmente, não está presente na IPBrick qualquer ferramenta de identificação de falhas. Ao invés, quando ocorre uma falha nos servidores IPBrick, é a equipa de

apoio a clientes que tem a responsabilidade de a solucionar manualmente. Este processo manual causa atrasos exagerados na resolução de problemas.

O objetivo desta dissertação é o desenvolvimento de um módulo IPBrick capaz de detetar falhas de forma a facilitar e automatizar o trabalho executado pela equipa de suporte e apoio a clientes. Esta ferramenta pode proporcionar uma maior brevidade na resolução de falhas, que seria uma mais valia importante para o sistema.

1.3 Estrutura da Dissertação

O presente documento encontra-se dividido em cinco capítulos. No primeiro capítulo, é introduzido o tema e destacada a importância do problema para a solução IPBrick. No segundo capítulo, são descritas as ferramentas com capacidade para solucionarem o problema. Apresenta-se ainda uma comparação entre as diversas soluções e os motivos que levam à escolha de uma solução em detrimento das outras, bem como os conceitos fundamentais para o tema. No terceiro capítulo, foca-se o trabalho prático desenvolvido. No quarto capítulo, demonstra-se o funcionamento da solução desenvolvida. Por fim, no quinto e último capítulo apresenta-se uma síntese do trabalho desenvolvido e sugerem-se possíveis melhorias com vista ao futuro desenvolvimento.

Capítulo 2

Estado da Arte

Neste capítulo definem-se os objetivos e requisitos que o produto final deverá respeitar, enquadra-se o tema na solução IPBrick e discutem-se as possíveis soluções encontradas, acompanhadas das suas vantagens e desvantagens.

2.1 Objetivos e Requisitos

Uma vez destacada a importância da deteção de anomalias nos servidores IPBrick é essencial perceber quais os objetivos e requisitos que devem ser cumpridos pelo produto a desenvolver.

2.1.1 Objetivos

Com as ferramentas disponíveis atualmente, quando um cliente reporta uma anomalia ao apoio técnico da IPBrick, são executados manualmente, um a um, um conjunto de procedimentos para despiste do problema. Esta situação pode provocar atrasos exagerados no processo de resolução.

O objetivo desta dissertação é desenvolver um módulo IPBrick que realize um *check-up* completo ao sistema, sempre que o administrador desejar e que, quando o processo terminar, um relatório com um resumo das anomalias encontradas seja apresentado numa interface *web* integrada no sistema IPBrick. O resultado deve ser suficientemente descritivo para agilizar a resolução do problema e identificar a sua possível origem.

2.1.2 Requisitos

Para atingir os objetivos, este módulo deve respeitar os seguintes requisitos:

- O administrador pode efetuar uma revisão total ao sistema e, como resultado, deve receber informação sobre o estado do *hardware* e *software* do servidor;
- Deve existir uma interface, integrada no sistema IPBrick, que permita ao administrador interagir com o módulo. Esta interface deve possibilitar o envio do comando de iniciar a análise ao sistema e, posteriormente, receber os resultados dessa análise;

- Caso sejam detetadas falhas, a descrição deve ser de tal modo pormenorizada, que identifique a origem do problema;
- O módulo deve ser estruturado de modo a prever futuras atualizações e atuar independentemente de outros módulos do sistema.

Relativamente aos requisitos de menor prioridade:

- O administrador pode beneficiar da opção de aumentar o nível de *debug*, isto é, pode conseguir detetar falhas mais ou menos graves de acordo com as definições pretendidas;
- Se a falha detetada possuir uma resolução conhecida, o módulo pode tentar efetuar alguns procedimentos com o objetivo de a solucionar;
- O módulo pode disponibilizar a opção de agendamento de análises ao sistema, como por exemplo o administrador poder ter a opção de agendar uma análise ao sistema todos os dias às horas por ele definidas.

2.2 Enquadramento

Seguidamente salienta-se a importância da deteção de falhas nos servidores e os procedimentos que devem ser seguidos para as detetar. É também apresentada uma breve descrição da empresa e do seu produto, focando o problema da deteção de falhas nos servidores IPBrick.

2.2.1 Importância da deteção de falhas

Na atualidade, verifica-se que em todas as organizações empresariais, a rede de comunicações assume um papel de elevada importância. É cada vez mais comum, estas organizações, possuírem servidores próprios que disponibilizam serviços como o correio eletrónico, o servidor *web* ou o VoIP. Desta forma, estas instituições garantem independência de outras para o fornecimento destes serviços, no entanto, ficam reféns de uma maior dependência da disponibilidade do seu servidor, uma vez que, se este falhar a comunicação interna e externa não é possível. Assim, descobrir a razão pela qual cada um dos serviços de um servidor não se encontra em correto funcionamento assume uma relevância extrema, principalmente para organizações que requerem padrões elevados de disponibilidade.

2.2.2 Procedimentos de deteção de falhas

O procedimento mais eficaz de detetar uma falha é efetuar testes que verifiquem o funcionamento dos parâmetros indispensáveis a cada serviço e.g. testar conetividade com o exterior, ou testar o registo de uma extensão no servidor VoIP . A este processo de verificação dá-se o nome de monitorização. Existem diversas ferramentas que cumprem esta função.

Toda a ferramenta de monitorização deve conter uma lista de procedimentos de execução, acompanhados da resposta esperada, visando o sucesso. Se a resposta recebida indicar sucesso, a

ferramenta mostra o correto funcionamento da entidade em questão. Se a resposta for diferente, é relatado o problema. As informações para o administrador do parque informático são exibidas numa interface *web*.

Por norma, à entidade responsável pela monitorização dá-se o nome de servidor de monitorização e às entidades monitorizadas o nome de *hosts* ou terminais. Por vezes, o teste não se restringe a uma comparação do valor esperado em caso de sucesso - valores *OK* - com o resultado obtido. Junta, a estes parâmetros, intervalos que definem valores sobre os quais o utilizador deve ter conhecimento, mas cuja importância não é demasiado elevada - valores *WARNING* - e intervalos que definem valores críticos para os quais o utilizador deve ser alertado imediatamente - valores *CRITICAL*.

2.2.3 IPBrick

A IPBrick SA [4] é uma empresa portuguesa, com sede no Porto, do ramo das telecomunicações. Do conjunto dos produtos por ela comercializados, destaca-se a IPBrick que consiste num servidor completo, tendo como base o sistema operativo Linux Debian [14], voltado para as organizações empresariais. As suas principais vantagens são a facilidade de utilização e de configuração. A disponibilidade de uma interface *web* limpa, intuitiva que permite fazer qualquer tipo de configuração de uma forma simples, mesmo para quem não possui grandes conhecimentos na área, contribui, também, para o seu sucesso.

2.2.4 Módulos IPBrick

Do ponto de vista estrutural, uma característica importante da IPBrick é a sua divisão em módulos. Estes operam de forma independente, ou seja, quando um módulo é executado não afeta, nem é afetado por outros. Estes módulos são componentes *plug-and-play* fornecedores de mais funcionalidades. Quando não instalados, o sistema não depende destes para funcionar.

Um módulo IPBrick é constituído essencialmente por dois componentes - a página *web* e a base de dados. A página *web* é responsável pela disponibilização de uma interface de configuração do módulo e pela apresentação dos resultados da execução ao utilizador do sistema. Esta interface *web* permite ao utilizador interagir com o sistema.

A base de dados armazena a informação relativa ao módulo. A informação contida na página *web* e todos os parâmetros relevantes encontram-se na base de dados de modo a facilitar futuras atualizações ou alterações. Esta base de dados tem uma estrutura independente de outros módulos e apenas fornece dados para o módulo em questão.

2.2.5 Módulos IPBrick.IC

A base da IPBrick é constituída por dois módulos fundamentais, a IPBrick.I [1] e a IPBrick.C [1] A IPBrick.I contém os serviços necessários para garantir a conetividade *Intranet* e a IPBrick.C garante os serviços de comunicação. Uma breve descrição de cada um destes módulos é apresentada nas secções seguintes.

2.2.5.1 IPBrick.I

A IPBrick.I tem a capacidade de garantir a conetividade interna da empresa. Na figura 2.1 podem observar-se alguns dos serviços disponibilizados, de que se destacam o servidor de *e-mail*, de ficheiros, de domínio e de impressão.

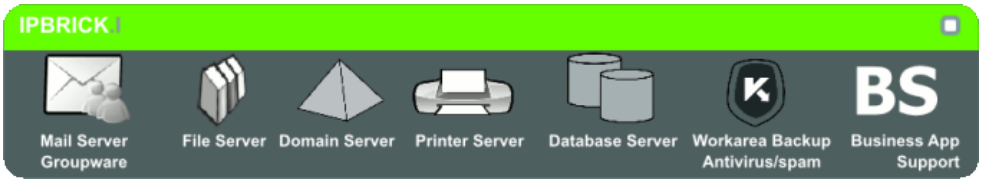


Figura 2.1: Serviços da IPBrick.I [1]

Um aspeto importante a referir é a escalabilidade oferecida por este módulo. No caso de existir mais do que um servidor IPBrick, é possível configurar a IPBrick.I [15] em três modos distintos [16] - *master*, *slave* e *Active Directory client* - com alterações ao nível do LDAP [17] e *Active Directory*:

- No modo *master* o servidor LDAP é a própria máquina;
- No modo *slave* o servidor LDAP é uma réplica sincronizada do servidor *master* configurado;
- No modo *Active Directory client* os serviços autenticam-se remotamente no servidor LDAP *master* configurado, sem que exista qualquer réplica no cliente.

Com estas soluções é possível distribuir os utilizadores por vários servidores, evitando a sobrecarga de um nó ou por qualquer outra razão que justifique uma separação entre grupos de *hosts*, sendo que a informação da localização das contas fica registada nos servidores LDAP.

2.2.5.2 IPBrick.C

O módulo *IPBrick.C* contém todos os serviços que permitem a comunicação com o exterior, com destaque para os serviços de *mail relay*, VoIP *gateway* e servidor *web*.

O comportamento correto consiste em reencaminhar para este módulo o tráfego de comunicações que tem como destino o exterior - tráfego VoIP, SMTP [18] ou Hypertext Transfer Protocol (HTTP) [19] . Neste processo verifica-se uma separação entre o tráfego interno e externo proporcionador de um maior grau de segurança.

Figura 2.2: Serviços da IPBrick.C [1]

Este módulo também atua como servidor Virtual Private Network (VPN), assumindo-se como mais uma ferramenta importante em muitas empresas. Outra mais valia é a integração de antivírus e *firewall* para controlar, quer o tráfego proveniente do exterior com destino interno, quer o tráfego interno com destino exterior. Estas duas ferramentas formam o primeiro escudo de segurança ao nível das comunicações.

2.2.6 Importância da deteção de falhas na IPBrick

A IPBrick está presente num elevado número de clientes que exigem padrões elevados de disponibilidade. Nestes clientes, como é sabido, uma falha dos serviços causa transtornos de elevadas proporções. Mesmo para clientes cujas necessidades de disponibilidade não sejam tão elevadas, a rápida resolução de qualquer problema é uma mais valia para o produto final. Sendo útil para todos, é sumamente útil para os servidores IPBrick com diversos clientes, onde as falhas exigem a rápida identificação da fonte do problema para posterior resolução.

Essencialmente, as falhas nos servidores IPBrick podem ser divididas em três grupos - conectividade com os *hosts*, conectividade com o exterior e falhas nos serviços.

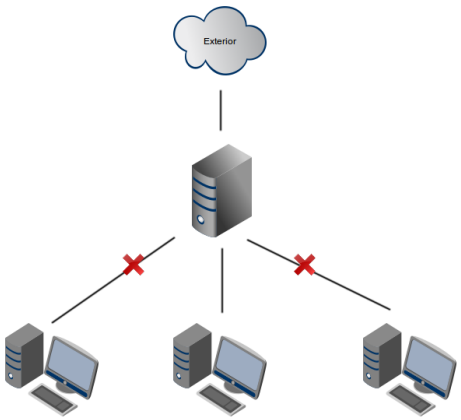


Figura 2.3: Falha na ligação com os *hosts*

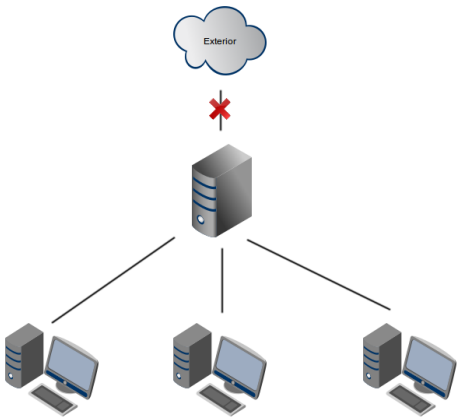


Figura 2.4: Falha na ligação com o exterior

2.2.6.1 Conectividade com os *hosts*

As falhas na conectividade com os *hosts* ocorrem quando as ligações entre uma ou mais máquinas da rede e o servidor são interrompidas. Na figura 2.3 pode observar-se um exemplo.

No centro da figura está representado o servidor IPBrick que faz a ligação da rede interna à externa. A rede interna é composta por três terminais ou *hosts*, mas dois destes não têm conectividade com o servidor, incapacitando o acesso aos recursos da rede. As anomalias deste género estão relacionadas com problemas da rede interna, configurações incorretas nos *hosts* ou nos servidores ou avarias de *hardware*.

2.2.6.2 Conetividade com o exterior

As falhas na conetividade com o exterior ocorrem quando a ligação do servidor ao exterior é interrompida. Na figura 2.4 pode observar-se um exemplo.

Tal como em 2.2.6.1, no centro da figura está representado o servidor IPBrick que faz a ligação da rede interna à externa e a rede interna também é constituída por três terminais. Neste caso, ao contrário da figura 2.3, todos os terminais têm conectividade com o servidor IPBrick, porém, a ligação do servidor para o exterior foi interrompida. As anomalias que originam esta situação deverão estar relacionadas com falhas na ligação ao Internet service provider (ISP) ou erros de configuração do servidor. No primeiro caso, pode dever-se a períodos de paragem programados para atualizações do sistema ou devido a erros inesperados por parte do fornecedor do serviço de *internet*. No segundo caso, pode dever-se à configuração errada dos parâmetros do servidor - e.g. rotas, endereços IP, entre outros factores - ou devido a equívocos de configuração da *firewall* ou antivírus - e.g. a *firewall* ou o antivírus podem estar a bloquear o tráfego externo devido a erros de configuração.

2.2.6.3 Falha nos serviços

As falhas nos serviços são as que causam mais transtornos ao nível dos servidores. Mesmo que exista conectividade interna e externa com o servidor, as falhas nos serviços provocam a falta de resposta aos pedidos e, por conseguinte, a falha no acesso aos recursos. Normalmente, estas anomalias resultam de erros na configuração ou falhas durante a execução do serviço. Neste segundo caso, as falhas podem dever-se ao facto do serviço não se encontrar em execução - e.g. inicialização de forma incorreta, paragem forçada por falta de recursos físicos - ou devido à incapacidade de resposta do serviço - e.g. sobrecarga de pedidos, falta de memória.

2.2.7 Serviços IPBrick

Depois de introduzido o tema, tem interesse aprofundar um pouco mais os serviços presentes na solução IPBrick, uma vez que são parte fundamental deste projeto.

Os serviços são aplicações que são executadas em *background*. Durante a sua execução são-lhe direccionados pedidos, os quais são processados e respondidos. As funcionalidades de servidor

são implementadas pelos serviços devido a esta capacidade de responder a pedidos específicos. e.g. a funcionalidade de servidor *web* é implementada por um serviço que processa os pedidos HTTP - o Apache [20].

O sistema operativo IPBrick inclui diversos serviços que fornecem à solução as funcionalidades essenciais de um servidor completo - e.g. Apache, PostgreSQL [21], OpenLDAP [22], entre outros. Nesta secção apresenta-se uma breve descrição desses serviços, destacando a sua importância para a solução e demonstrando uma perspetiva geral dos conceitos básicos necessários para a compreensão do trabalho.

2.2.7.1 Sistema

Apesar de não se tratar de um serviço, contextualizando no tema da dissertação. os componentes básicos do sistema podem causar problemas em vários serviços. Entendam-se por componentes básicos do sistema três parâmetros considerados fundamentais para a execução saudável dos serviços: o disco rígido, a capacidade do processador e a memória Random Access Memory (RAM). Nenhum serviço poderá ser executado se não existirem capacidades do processador, da memória RAM e do disco rígido que sustentem as necessidades. Na solução IPBrick, tal como em todos os sistemas operativos, é importante obter métricas fornecedoras da informação acerca do comportamento do sistema, permitindo, assim, obter conhecimento dos serviços que estão a ser afetados ou que podem vir a ser afetados pela escassez de recursos disponíveis.

2.2.7.2 Base de dados

O incremento das necessidades de armazenamento potenciou o surgimento de estruturas organizadas de dados, as bases de dados, que são frequentemente utilizadas em situações que implicam um relacionamento entre conjuntos de dados e a gestão eficiente dos mesmos. Numa base de dados, a informação é organizada em tabelas relacionadas entre si. As tabelas executam o agrupamento dos dados em linhas e colunas. As linhas,designadas tuplos, representam um novo objeto guardado e as colunas representam os atributos referentes a esse objeto.

Um sistema de gestão de base de dados tem a responsabilidade de gerir o acesso às tabelas e, por conseguinte, aos dados. A comunicação com estes sistemas é realizada através de pedidos - a expressão correta em inglês é *queries* - dentro de um sintaxe específica. Apesar de existirem exceções à regra, maioritariamente, os sistemas de gestão de bases de dados requerem que as *queries* utilizem uma sintaxe comum definida pela linguagem Structured Query Languag (SQL) [23]. Existem diversos sistemas de gestão de base de dados SQL entre os quais se destacam: PostgreSQL e MySQL [24]. Na solução IPBrick, o serviço que implementa o sistema de gestão de base de dados é o PostgreSQL. O PostgreSQL é um serviço *open-source* que resultou de uma evolução constante ao longo de vários anos do projeto Ingres [25]. Presentemente, é visto como um sistema robusto e muito eficaz e, por isso, tornou-se no sistema de base de dados mais utilizado. O seu licenciamento, que confere um grau elevado de liberdade aos programadores/utilizadores, contribuiu muito para o seu sucesso.

O sistema de gestão de base de dados assume um papel fundamental no desempenho de toda a solução IPBrick, como tal, apresenta-se como um serviço crítico de que todos os outros serviços, bem como o próprio funcionamento do sistema operativo, dependem. Para que a importância deste serviço fique bem clara convém perceber melhor o funcionamento da solução IPBrick. Esta solução fornece ao utilizador uma interface *web* para a gestão completa do servidor. Nessa interface *web* existem inúmeras configurações que podem ser alteradas. Obviamente, não faria sentido que, no momento em que cada parâmetro fosse alterado, houvesse uma comunicação com o servidor para informar a modificação. Se assim fosse, podiam surgir facilmente problemas de incompatibilidades nos serviços, devido a erros nas configurações.

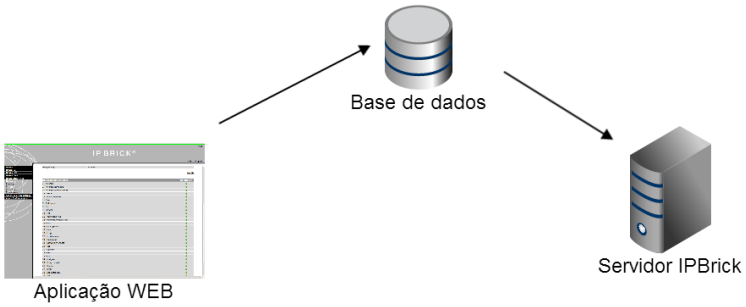


Figura 2.5: Processo de alteração de configurações no sistema operativo IPBrick.

Para evitar estes inconvenientes, a comunicação da interface *web* com o sistema operativo e os seus serviços é realizada através da base de dados. As alterações efetuadas na interface *web* não são efetuadas diretamente no sistema operativo. A informação segue o fluxo representado na figura 2.5, isto é, os dados são guardados na base de dados e, só quando o utilizador seleccionar a opção *"Apply Configurations"*, no menu concebido para o efeito, é que são realizadas as alterações no sistema operativo. Este passo intermédio, para além de evitar problemas de incompatibilidade, permite o registo de todas as alterações, facilitando a recuperação em caso de insucesso nas operações realizadas.

2.2.7.3 DNS

O DNS [26] é um sistema de gestão de nomes cuja principal função é a conversão de endereços IP em nomes e vice-versa. Em qualquer rede de média ou grande dimensão torna-se indispensável a existência deste sistema para evitar o esforço necessário para memorizar todos os endereços IP da rede.

Os servidores DNS operam sobre o modelo cliente-servidor. A sua estrutura evidencia uma organização hierárquica que se comporta como uma base de dados com informação sobre os nós de uma rede IP, sendo essa informação organizada em domínios. A notação usada para a atribuição de nomes é Fully Qualified Domain Name (FQDN):

E.g girigiri.gbrmpa.gov.au

Neste exemplo, "girigiri"representa o nome da máquina e "gbrmpa.gov.au"representa o domínio em que está inserido. O domínio é definido com três níveis, o domínio de topo e os 2º e 3º níveis, mas o sistema não obriga a um número fixo de níveis. Na figura 2.6 é possível observar as sucessivas consultas efetuadas no processo de resolução deste nome. No decorrer do mesmo processo o endereço é lido de trás para a frente, ou seja, desde o domínio mais abrangente até ao menos abrangente. A gestão dos domínios de topo é realizada pela Internet Corporation for Assigned Names and Numbers (ICANN) [2].

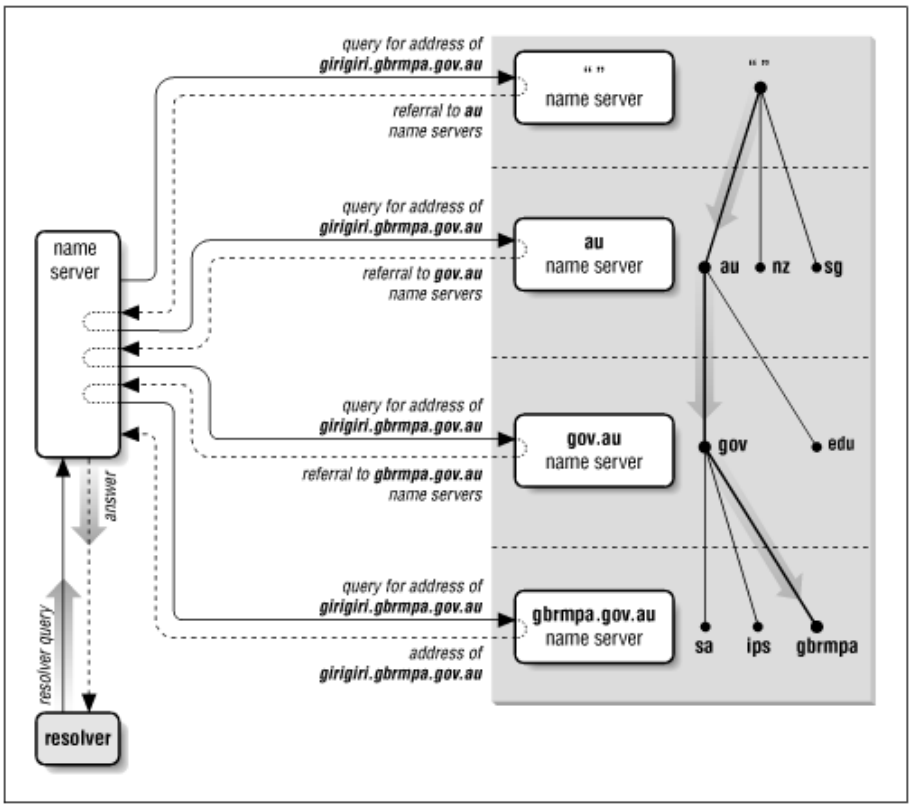


Figura 2.6: Arquitetura hierárquica dos servidores DNS [2]

Um servidor de DNS é responsável pela gestão dos registos de uma determinada parte do domínio, que normalmente se designa por zona.

Pode analisar-se a função de um servidor DNS segundo duas perspetivas:

- Servidor DNS interno - Um servidor DNS interno resolve nomes de uma rede privada. Este tipo de servidores encontram-se numa Local area network (LAN) privada e só respondem aos pedidos dessa rede;
- Servidor DNS público - Um servidor DNS público resolve apenas nomes que pertençam a IP públicos ou bem conhecidos. Normalmente o servidor DNS público de um domínio é gerido pelo ISP ou uma organização específica.

O servidor de DNS permite também realizar a tarefa inversa, ou seja, a resolução de endereços IP em nomes FQDN.

Numa solução como a IPBrick, o servidor DNS assume um papel muito relevante. Com a quantidade de máquinas geridas pela solução torna-se fundamental uma gestão eficiente dos nomes e endereços IP. Na IPBrick, o servidor DNS é implementado pelo serviço bind9 [27]. A gestão do domínio público pode ser efetuada pelos ISP, mas, a gestão de todos os subdomínios e máquinas é assegurada por este serviço.

2.2.7.4 DHCP

Para que a comunicação entre dois quaisquer dispositivos de uma rede se torne possível, é necessário que ambos possuam um endereço IP válido dessa rede. Numa analogia simples, o endereço IP pode ser comparado ao endereço de uma residência. Para enviar uma carta para uma pessoa, é obrigatório saber a morada de destino, caso contrário não será entregue no local pretendido. De igual modo, numa comunicação entre máquinas, é necessário saber para onde enviar os pacotes. Por este motivo é essencial que todos as máquinas da rede possuam um endereço IP para poderem contactar e ser contactadas por outras.

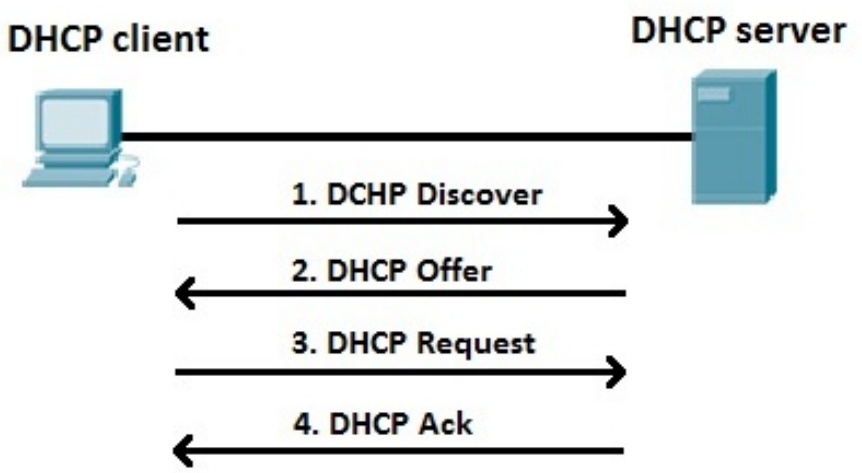


Figura 2.7: Processo de atribuição de um endereço IP [3]

O DHCP [28] é o serviço responsável pela atribuição automática de endereços IP na rede. O seu funcionamento baseia-se num modelo cliente-servidor. Quando o cliente é ligado à rede e necessita de um endereço IP realiza um pedido *broadcast* à procura de servidores DHCP. Se houver algum servidor disponível, este responde com o IP, a *default gateway* e o endereço do servidor DNS sugerido para aquele cliente. Quando o cliente recebe esta informação responde indicando que aceitou aquelas indicações. Por fim, o servidor encerra a ligação com um Acknowledgement (ACK). O sistema operativo IPBrick nem sempre é definido como o servidor DHCP para toda a rede, mas no caso de o ser, a sua importância nessa rede pode ser facilmente demonstrada com um breve exemplo. Imagine-se uma rede com 1000 dispositivos, seria uma tarefa difícil

tentar perceber quais os endereços IP que ainda não estão a ser usados. Um servidor DHCP executa de forma automática essa tarefa e ainda permite a configuração automática da informação de *gateway*, máscara de rede e servidores DNS. Com este serviço, um dispositivo ligado à rede obtém acesso automático sem necessidade de configuração manual.

A gestão centralizada de IP permite, ainda, ao servidor DHCP a atribuição de IP fixos em determinadas máquinas, através do *Mac Address*. Esta é uma funcionalidade importante em locais cujo acesso à rede seja limitado às máquinas conhecidas, mesmo que o meio de transmissão não se encontre limitado a estas. A título de exemplo, numa instituição bancária é inconveniente que qualquer pessoa possa ligar-se à rede interna.

2.2.7.5 LDAP

Com a crescente utilização de dispositivos ligados em rede, não só computadores, mas também impressoras ou telefones, surgiu uma necessidade de evitar o armazenamento local da informação de utilizadores, contas, serviços e aplicações, de forma a possibilitar a independência dos recursos físicos. Um servidor LDAP permite a centralização de todas estas informações em um ou vários servidores.

E.g. com um servidor LDAP configurado, um utilizador pode fazer *login* em qualquer máquina da rede e terá sempre disponível a mesma configuração, os mesmos ficheiros e as mesmas aplicações da sua conta remota. Sem um servidor LDAP, os dados da sua conta estariam apenas disponíveis em uma máquina.

O LDAP é o protocolo que define o método de acesso aos dados do utilizador armazenados no servidor. Os servidores LDAP utilizam o protocolo LDAP para a troca de informação. As funcionalidades de servidor LDAP são implementadas na IPBrick pelo serviço openLDAP [22]. A importância deste serviço para a solução IPBrick está diretamente relacionada com a estratégia da empresa.

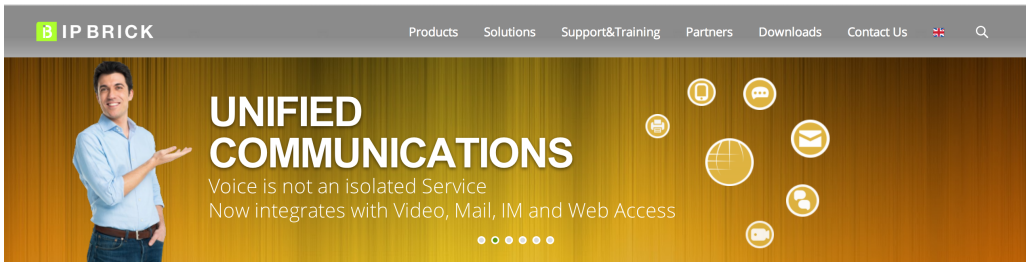


Figura 2.8: Destaque UCoIP no site www.ipbrick.com [4]

A funcionalidade presente na solução IPBrick que se tem evidenciado como fundamental na sua estrarégia é o Unified Communications over IP (UCoIP). Este conceito é apresentado como um *slogan* da empresa , como pode observar-se na figura 2.8 , e é uma das razões que vem contribuindo para o sucesso do sistema operativo.

Para facilitar a compreensão da definição de UCoIP, suponha-se esta situação vulgar nos dias que correm. Um trabalhador de uma determinada organização possui vários meios pelos quais

pode ser contactado, a extensão telefónica interna, o endereço de *e-mail*, o endereço *skype*, o número de telemóvel, o endereço do IM, entre outros. Será necessário um colaborador ter que saber todos estes endereços e números? Com outras soluções é. O conceito UCoIP resolve esta ineficiência permitindo que cada utilizador seja apenas identificado pelo seu endereço de *e-mail*. E.g. Se for necessário contactar o utilizador "user1" da empresa cujo domínio é "ipbrick.com", sabe-se que é possível contactá-lo utilizando apenas o endereço user1@ipbrick.com, qualquer que seja o meio de comunicação utilizado. É possível enviar um *e-mail* para este endereço, contactar pelo IM o mesmo endereço, telefonar para a extensão do utilizador "user1" através do mesmo endereço e, caso não esteja disponível, a chamada será reencaminhada para seu o telemóvel. Portanto, o utilizador é reconhecido em qualquer serviço do sistema através do seu endereço de *e-mail* sem necessidade de inúmeros contactos diferentes. Esta mudança também traz melhorias ao próprio utilizador porque evita a utilização de nomes de utilizador e palavras-passe diferentes para os diversos serviços.

A associação do utilizador em diversos serviços com apenas um endereço, é possível devido à informação guardada no servidor LDAP. Por isso, este serviço é tão importante para o funcionamento da solução.

2.2.7.6 Firewall

A *firewall* é a componente responsável pela regularização do tráfego de rede de entrada e saída.

A cada serviço que necessita de comunicação com o exterior da máquina é atribuída uma porta de comunicação, para que, o emissor envie a informação pela porta que o recetor escuta. A gestão de tráfego efetuada pela *firewall* é baseada em regras aplicadas ao tráfego das diferentes portas.

Dada a importância do tráfego que flui na rede interna de uma organização empresarial comportando informações confidenciais que não devem ser divulgadas, a gestão eficiente e exclusiva deste tráfego é uma mais valia. A *firewall* constitui uma primeira barreira para todo ele e, por essa razão, assume um papel de destaque no campo da segurança.

Na sistema operativo IPBrick, as funcionalidades de *firewall* são implementadas pela ferramenta IPTables [29]. Pelo facto dos servidores IPBrick, na maioria das vezes, efetuarem a ligação da rede interna com a rede externa, ou seja, serem o ponto de entrada e saída da rede, a sua *firewall* é o filtro principal, não só para os próprios servidores mas também para toda a rede.

2.2.7.7 VoIP

A tecnologia VoIP consiste, como o próprio nome indica, na transmissão de voz sobre redes IP. Esta tecnologia surgiu quando ficou clara a capacidade emergente das redes IP para transporte de dados. Quando comparada com a rede telefónica convencional Public Switched Telephone Network (PSTN), é indubitavelmente mais competitiva em termos de custo e de capacidade de implementação de novas funcionalidades.

Com a vulgarização do conceito VoIP surgiram serviços implementadores das funcionalidades de Private Branch Exchange (PBX), denominados Internet Protocol Private Branch Exchange (IPPBX), para serem distinguidos dos convencionais. O Asterisk [30] é o serviço mais reconhecido nesta área, não só porque implementa funcionalidades que não existiam nos sistemas de telefonia convencional, mas, também, porque o código fonte pode ser alterado e adaptado às necessidades dos utilizadores.

O modo de funcionamento do Asterisk baseia-se em quatro conceitos fundamentais [5]:

- Extensão - uma extensão refere-se ao número identificador do chamador [31]. Por norma, cada telefone tem uma extensão associada - e.g. Quando se efetua uma chamada para o número 123, está a ser efetuada uma chamada para a extensão 123;
- Canais - os canais representam uma ligação, semelhante à figura 2.9, entre o sistema Asterisk e um terminal telefónico físico ou virtual;

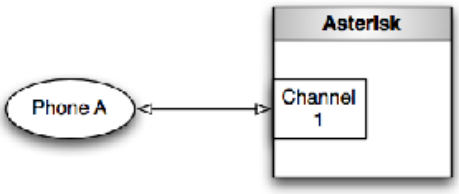


Figura 2.9: Exemplo de um canal Asterisk [5]

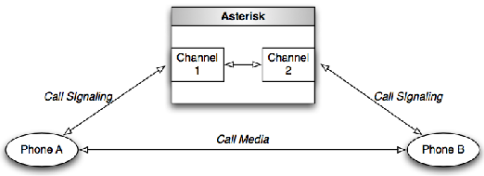


Figura 2.10: Ligação entre dois canais Asterisk [5]

- Ligação de canais (*Channel Bridging*) - a existência de canais isolados não faculta qualquer vantagem para o utilizador. Diferentemente, é a ligação entre dois ou mais canais que serve de base a todas as funcionalidades do Asterisk. A realização de uma chamada não é mais do que a ligação entre dois canais, semelhante à representada na figura 2.10 - o estabelecimento de uma ponte, daí provém o termo em inglês *channel bridging*;
- Contexto - define um conjunto de extensões e as regras que as caracterizam, como por exemplo, as rotas, a segurança, a autenticação, entre outras [31]. As rotas são as regras que regem o estabelecimento de chamadas. É com base nestas que a chamada é reencaminhada para o destino correto;
- *Dial-plan* - define um conjunto de contextos, englobando todos os conceitos anteriores [32]. É, por isso, o ponto central do serviço Asterisk. A configuração deste serviço baseia-se na configuração do *dial-plan*.

Para o utilizador receber ou efetuar chamadas, deve realizar o registo do seu terminal de forma a que lhe seja atribuído um canal e uma extensão. Todos os processos que envolvem comunicação como os de registo e de realização e receção de chamadas, implicam uma troca de mensagens entre o terminal e o IPPBX. A este método de troca de mensagens, que não implica uma transferência

efetiva de dados *audio* ou *video*, dá-se o nome de sinalização. O protocolo mais utilizado nas mensagens de sinalização é o protocolo Session Initiation Protocol (SIP).

Já aqui foi descrita a importância do UCoIP para a IPBrick. Este serviço de VoIP é mais um serviço englobado nesse conceito de comunicações unificadas sobre IP. A voz sobre IP é uma funcionalidade cada vez mais utilizada e que concede enormes vantagens para o utilizador, quer ao nível de *performance*, quer ao nível de custo.

Os serviços responsáveis pela implementação das funcionalidades VoIP na IPBrick são o Asterisk e o Kamailio [33]. Os conceitos fundamentais do Asterisk já foram introduzidos na secção anterior. Clarifica-se, agora, a função do Kamailio.

O Kamailio executa as funções de *proxy* SIP, sendo da responsabilidade deste serviço realizar a interligação do servidor Asterisk com a rede. As mensagens de sinalização são primeiro processadas no Kamailio e só depois reencaminhadas para o Asterisk. Esta centralização da gestão eleva o nível de segurança porque o *proxy* atua como primeiro filtro e garante escalabilidade porque é possível distribuir a carga por diversos IPPBX recorrendo a um *proxy*.

Atualmente, é possível construir uma solução de PBX baseada no Asterisk e Kamailio com as mesmas ou até mais funcionalidades, quando comparada com uma solução de PBX tradicional e com um orçamento muito mais competitivo.

Com a facilidade proporcionada ao utilizador na hora de configurar todo o sistema operativo através da interface *web*, por vezes, ocorrem erros de configuração difíceis de detetar ou até completamente despercebidos, apenas detetados em *runtime* . No Asterisk e Kamailio esses erros são relativamente frequentes e, por isso, este projeto terá grande impacto neste serviço.

2.2.7.8 Correio eletrónico

O correio eletrónico vem-se destacando como meio de comunicação de eleição por parte das organizações empresariais. Hoje em dia, é cada vez menos comum uma empresa não fazer uso do *e-mail*, quer para comunicação externa, quer interna.

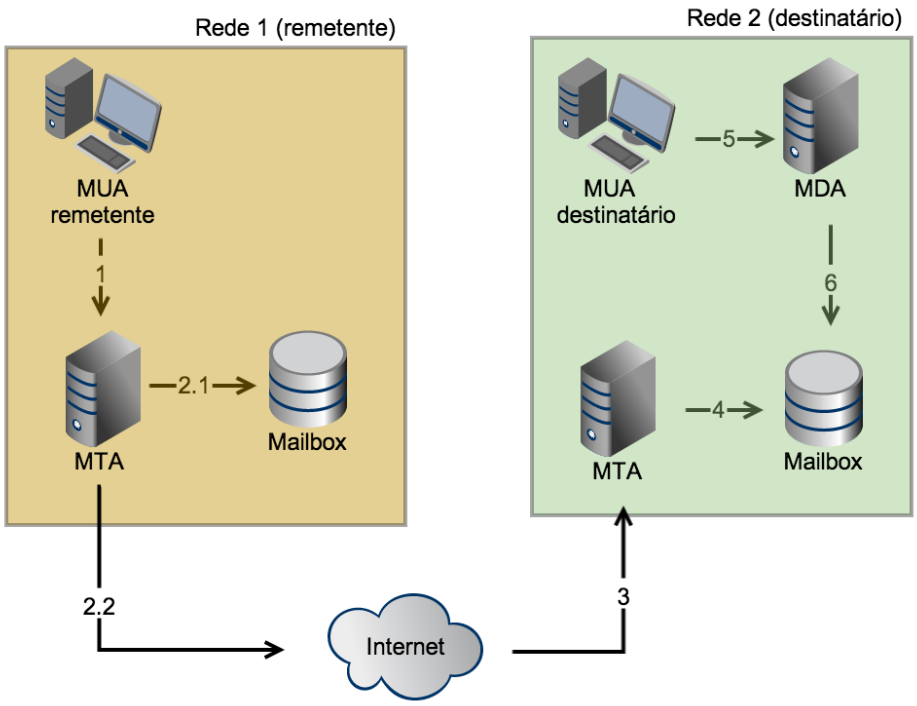


Figura 2.11: Entidades intervenientes no processo de envio e receção de e-mails

O modelo habitual para a configuração do serviço de email assemelha-se ao representado na figura 2.11 . Nesta figura é possível observar as quatro entidades intervenientes no processo de troca de mensagens de *e-mail* [6]:

- O Mail User Agent (MUA) é a ferramenta que permite ao utilizador enviar e receber correio eletrónico e.g. Thunderbird [34], Outlook [35];
- O Mail Transfer Agent (MTA) permite a troca de mensagens de correio eletrónico com outros MTA. Todos os MUA de uma rede enviam os seus *e-mails* através de um MTA e apenas este comunica com outras redes;
- O Mail Delivery Agent (MDA) permite a recepção do correio eletrónico;
- A *Mailbox* permite guardar as mensagens de *e-mail*.

Na mesma figura é ainda possível seguir o trajeto de uma mensagem de *e-mail* desde o remetente ao destinatário:

- **1.** O utilizador escreve e envia a mensagem no seu MUA e este envia-a para o MTA configurado;
- **2**. O MTA processa a mensagem e executa uma das duas tarefas **2.1.** ou **2.2.**;
- **2.1.** Se o destinatário da mensagem pertencer à mesma rede do remetente, o MTA guarda a mensagem na *mailbox* local;

- **2.2.** e **3.** Se o destinatário da mensagem pertencer a uma rede diferente do remetente, o MTA envia-a para o MTA da rede do destinatário;
- **4.** O MTA da rede do destinatário guarda a mensagem na *mailbox* configurada;
- **5.** Quando o destinatário aceder à sua caixa do correio no seu MUA, é efetuada uma consulta ao MDA que lhe devolve os *e-mails* presentes na *mailbox*.

Uma vez introduzidos os conhecimentos básicos da estrutura do serviço de *e-mail*, atente-se aos protocolos utilizados. Essencialmente, existem dois tipos de protocolos que intervêm no processo de troca de *e-mails* - os protocolos de envio de mensagens e os protocolos de recepção de mensagens . O protocolo de envio de mensagens mais utilizado é o SMTP e os protocolos de recepção mais comuns são o POP e o IMAP. Na figura 2.12 está representada uma situação comum na qual o MTA e MDA partilham a mesma máquina física, aqui designada *Mail Server*. Observa-se que os protocolos POP e IMAP se restringem à recepção de *e-mails*, isto é, são usados, apenas, na comunicação efetuada entre o MDA e o MUA (passo **5** da figura 2.11) . Já o protocolo SMTP é usado no envio de mensagens do MUA para o MTA (passo **1** da figura 2.11) e também na troca de mensagens entre MTA (passos **2.2** e **3** da figura 2.11).

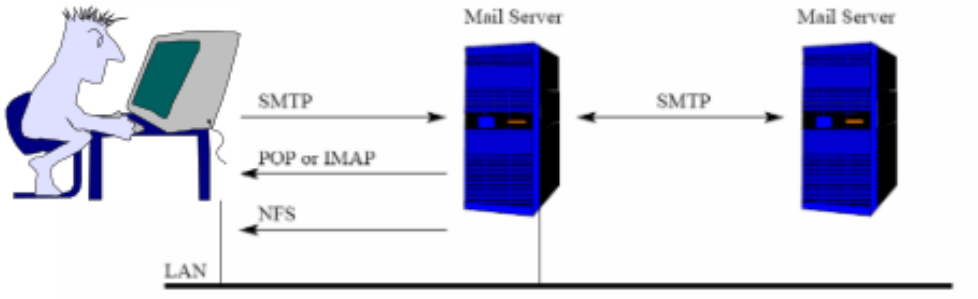


Figura 2.12: Utilização dos protocolos POP, IMAP e SMTP [6]

Embora o SMTP seja unanimemente aceite como o protocolo de referência para o envio de *e-mails*, a situação não se repete nos protocolos de recepção. Aí não existe um protocolo de referência, existem dois, o POP e o IMAP. Como ambos são universalmente utilizados, impõe-se que os servidores possuam compatibilidade com os dois.

Ao início pode parecer estranho que dois protocolos que realizam a mesma tarefa sejam utilizados numa proporção semelhante, mas, olhando para as características de cada um, rapidamente se percebe que os dois realizam as mesmas tarefas de forma diferente e que ambos são necessários.

POP e IMAP representam dois paradigmas diferentes na abordagem à recepção de *e-mail* - o protocolo POP segue o paradigma *offline* e o protocolo IMAP segue o paradigma *online*. Tal como o nome sugere, o paradigma *online* impõe uma constante ligação à rede e o paradigma *offline* é mais adequado para situações com limitações de tempo de ligação. Com o protocolo IMAP, o MUA apenas consulta a caixa de correio remota do utilizador, e descarrega, temporariamente, apenas as mensagens que se pretendem ler. Já, com o protocolo POP, o MUA descarrega todas as

mensagens presentes na caixa de correio remota para a sua caixa de correio local, não existindo armazenamento remoto. Na tabela da figura 2.13 é efetuada uma comparação entre os dois protocolos e são resumidas as características principais, já aqui descritas. A IPBrick, como prestadora

	POP3	IMAP
RFC que o descreve	1939	3501
Port TCP usado	110	143
Port TCP para SSL	995	993
Arquivo do e-mail	Sistema do cliente	Servidor
Leitura	Offline	Online
Tempo de ligação	Reduzido	Muito
Utilização dos recursos do servidor	Mínima	Intensa
Mailboxes multiplas	Não	Sim
Recomendado para utilizadores móveis	Não	Sim
Controlo do download pelo utilizador	Reduzido	Muito
Download parcial das mensagens	Não	Sim
Complexidade da implementação	Não	Sim

Figura 2.13: Comparação entre os protocolos POP e IMAP [6]

de serviços e fornecedora de produtos para organizações empresariais, propõe uma solução que integra o serviço completo de envio e recepção de correio eletrónico no seu contexto UCoIP. A solução IPBrick contempla um servidor SMTP, o qmail [36], e os servidores POP, o courier-pop [37], e IMAP, o courier-imap [37]. Existem ainda mais dois serviços que completam a oferta de funcionalidades nesta área - o courier-pop-ssl [37] e o courier-imap-ssl [37]. Estes dois serviços proporcionam a recepção de *e-mails* sobre túneis Secure Sockets Layer (SSL) [38] que conferem maior segurança ao utilizador, habitualmente designados Post Office Protocol Secure (POPs) e Internet Message Access Protocol Secure (IMAPs). Note-se que o qmail também prevê a utilização de SMTP sobre túneis SSL, geralmente designado Simple Mail Transfer Protocol Secure (SMTPs).

Contextualizando estes serviços na estrutura apresentada na secção **??**, o Qmail executa as funções de MTA, os serviços courier executam as funções de MDA.

2.2.7.9 IM

O IM é um serviço que permite a troca instantânea de mensagens escritas entre dois ou mais utilizadores. É ,por isso, uma ferramenta muito utilizada em organizações empresariais. No sis-

tema IPBrick, o IM insere-se no conceito UCoIP e, por esta razão, também é um dos serviços base de comunicação. O serviço que implementa as funcionalidades de IM é o Ejabberd [39], complementado com algumas funcionalidades extra, desenvolvidas pela IPBrick, com destaque para a possibilidade de gravação das mensagens trocadas.

2.2.7.10 Servidor web

Um servidor *web* consiste num módulo responsável pelo alojamento de paginas *web* que permitem a distribuição de conteúdo através da Internet ou da rede local.

O servidor *web* mais popular é o Apache. O Apache é compatível com quase todos os sistemas operativos e teve um papel importante nos primeiros passos da World Wide Web (WWW) devido à sua estabilidade e escalabilidade. Hoje em dia é o servido *web* mais usado a nível mundial. O seu funcionamento baseia-se nas respostas aos pedidos HTTP dos clientes, respostas estas que também estão sujeitas às regras do protocolo HTTP. De entre as múltiplas funcionalidades que o servidor Apache disponibiliza, destaca-se a capacidade de oferecer várias páginas *web* através do mesmo servidor. Isto é possível devido à existência de instâncias virtuais conhecidas por VirtualHosts. Para o cliente, a existência de VirtualHosts é semelhante à existência de vários servidores *web* a correrem simultaneamente na mesma máquina. Mas como a coexistência de vários servidores *web* na mesma máquina não é possível porque não é permitido que dois ou mais serviços respondam na mesma porta ao mesmo tempo, o Apache ultrapassou esta limitação com a criação de instâncias virtuais que respondem a diferentes pedidos HTTP, distinguindo-os através da porta ou do nome da página *web*.

A solução IPBrick baseia-se na simplicidade conferida ao utilizador no momento de configurar o servidor através da interface *web*. Porém, a interface *web*, para estar disponível, necessita de um servidor *web*. O Apache, neste campo, usado como servidor *web* na IPBrick, é um dos serviços mais importantes por facultar a mais valia do sistema, a facilidade de configuração.

2.2.7.11 Fax

O servidor Fax é um serviço que permite o envio de *faxes* sem recorrer à digitalização, isto é, permite o envio de *faxes* com recurso apenas a *software*.

O serviço que implementa as funcionalidades Fax na IPBrick é o HylaFAX [40], complementado com as funcionalidades de FAX2MAIL e MAIL2FAX, permitindo enviar e receber *faxes* como mensagens de correio eletrónico.

Por se tratar de um serviço de comunicação também é algo que representa importância para o sistema IPBrick, e por isso lhe é dado este destaque.

2.2.7.12 Proxy

Um servidor *proxy* atua como intermediário na comunicação entre os clientes e os servidores. Como se pode observar na imagem 2.14, em que está representada a situação específica de um servidor *web*, o tráfego atravessa sempre o *proxy* antes de atingir o servidor *web*. Esta configuração

tem vantagens ao nível de *performance* porque o servidor *proxy* armazena os últimos pedidos na memória *cache*, evitando a constante consulta ao servidor *web*.

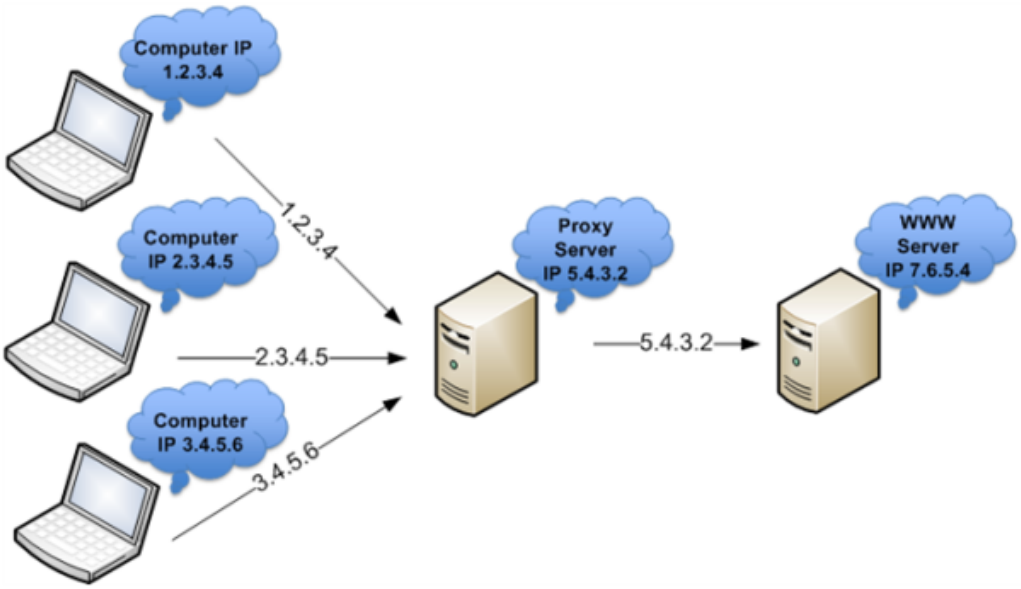


Figura 2.14: Exemplo de utilização de um servidor *proxy* para o tráfego *web* [7]

Os servidores *proxy* podem atuar em um de três modos distintos [41]:

- Normal - é necessário que os clientes sejam configurados para todo o tráfego fluir pelo servidor *proxy*;
- Transparente - o tráfego é reencaminhado para o servidor *proxy* sem que o utilizador tenha conhecimento. Todo o processo de reencaminhamento é impercetível para o utilizador;
- Com autenticação - o tráfego é desviado para o *proxy* mas só é reencaminhado para o destino se o utilizador estiver autenticado e devidamente reconhecido. Este método permite identificar e responsabilizar os utilizadores pelos seus atos na rede.

O serviço que implementa as funcionalidades de *proxy* no sistema operativo IPBrick é o Squid [42]. Entre as vantagens de utilizar um servidor *proxy* numa rede destacam-se a melhoria da experiência de acesso aos conteúdos da *internet* devido à memória *cache* e ao aumento do grau de segurança e de filtragem de conteúdos. Para os clientes IPBrick, a possibilidade de proporcionar estas vantagens aos seus colaboradores eleva a importância deste serviço no contexto da solução completa.

2.3 Soluções

Para a resolução do problema proposto foram pensadas, essencialmente, duas alternativas - desenvolvimento de um módulo IPBrick de raiz ou utilização de uma ferramenta de monitorização já existente. De entre as diversas ferramentas existentes optou-se por analisar em detalhe o Nagios pelos motivos que serão apresentados na secção 2.3.1.

2.3.1 Nagios

Normalmente, as ferramentas de monitorização executam *scripts* periodicamente que verificam o correto funcionamento dos serviços pretendidos.

O Nagios é uma ferramenta de monitorização *open-source* capaz de monitorizar terminais ou servidores. Tem a capacidade de registar estatísticas de recursos físicos - e.g. uso do disco rígido, uso de memória, uso do processador - e serviços de rede - e.g. verificar se os serviços se encontram em execução. Também é possível estruturar a rede hierarquicamente, ou seja, identificar a cadeia hierárquica entre pais e filhos e apresentá-la em rede sob a forma de árvore.

Uma característica que distingue o Nagios de outras ferramentas de monitorização é a possibilidade de serem desenvolvidos *plugins* para acrescentar funcionalidades. Assim, esta ferramenta torna-se muito poderosa, porque dá liberdade aos programadores para adicionar funcionalidades extra. Esta é a principal razão de ser apresentada, neste documento, uma análise mais detalhada ao Nagios e não outras ferramentas como o Zabbix [43], Zenoss [44] ou outras semelhantes.

Pelas suas mais valias, o Nagios tornou-se uma das ferramentas mais usadas neste setor e tem vindo a conquistar o interesse de comunidades de programadores [45] que desenvolvem *plugins*. Devido a este interesse, são já muitos os *plugins* disponíveis no *site* oficial [46].

2.3.1.1 História

O Nagios foi desenvolvido em 1996 [47] por Ethan Galstad e o seu primeiro nome foi Netsaint. Surgiu como uma aplicação para MicroSoft Disk Operating System (MS-DOS) que executava *ping* para verificar o correto funcionamento dos servidores. Posteriormente, Galstad optou por desenvolver a mesma aplicação, com algumas melhorias, para o Linux devido à maior facilidade de interação com os serviços de um servidor. Desde então, este *software* tem vindo a alargar significativamente as suas funcionalidades, o que lhe tem merecido diversos prémios, distinguindo-o como um dos melhores desta área.

2.3.1.2 Arquitetura

O Nagios é uma aplicação que segue a estrutura servidor-cliente. O servidor é responsável pelo processo de monitorização e o cliente é a máquina ou serviço que é monitorizado. A monitorização pode ser efetuada em serviços - e.g. servidor SMTP, servidor Post Office Protocol v3 (POP3), servidor IMAP, servidor HTTP - ou em *hosts* - e.g. servidores, *routers*, *switches*, estações de trabalho, impressoras. O Nagios distingue-se de outras ferramentas similares por não utilizar rotinas internas para verificar o estado dos serviços ou *hosts*. Quem é responsável por essa verificação são os *plugins* [8].

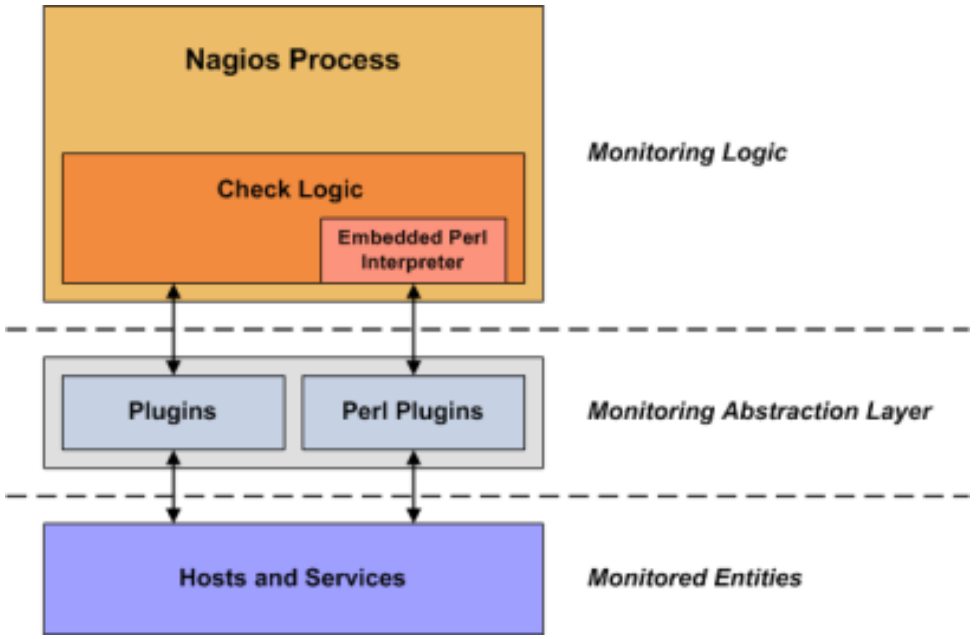


Figura 2.15: Arquitectura do Nagios [8]

Como se pode observar na figura 2.15, a arquitetura Nagios divide-se em três camadas distintas:

- A lógica de monitorização - responsável por executar os *plugins*, verificar o *output* e mostrar a informação organizada numa interface *web*. Não é da responsabilidade desta camada executar procedimentos diretamente. A sua função é invocar o *plugin* correto e analisar o resultado;

- Os *plugins* - responsáveis por executar um procedimento e retornar um valor que deve indicar o estado do serviço ou *host* a monitorizar. A camada em que estão inseridos fornece um nível elevado de abstração no sentido de ser possível monitorizar qualquer tipo de equipamento sem haver preocupação com outras camadas. A única preocupação deve ser o *output* do comando executado no *plugin*. Assim, é possível monitorizar não só qualquer tipo de *host* ou serviço mas também algo menos comum como a temperatura de uma casa ou mesmo um eletrodoméstico, desde que seja possível executar procedimentos para obter esses valores. De acordo com a documentação oficial [13], o valor retornado pelos *plugins* é analisado segundo a tabela 2.1. Pode ser escolhida qualquer linguagem para o desenvolvimento dos *plugins* desde que seja respeitado o *output* e.g. Perl [48], Shell Script, PHP [49] ou ficheiros já compilados de outras linguagens como C ou C++ [50].

Tabela 2.1: Resultados esperados na execução de um *plugin* [13]

Resultado esperado	Estado do serviço	Descrição
0	OK	O *plugin* conseguiu verificar o serviço e, aperentemente, está a funcionar corretamente.
1	Warning	O *plugin* conseguiu verificar o serviço mas foi registado uma ligeiro desvio relativamente ao resultado esperado.
2	Critical	O *plugin* detetou que o serviço pode não estar a ser executado ou foi registado um erro grave.
3	Unknown	Não foi possível verificar o estado do serviço. Provavelmente a ligação remota não foi possível.

Os *plugins* devem imprimir uma mensagem descritiva do erro que está a ocorrer, para facilitar a sua identificação. De facto, os *plugins* são *scripts* de execução respeitadores apenas de duas regras - o valor retornado deve ser de acordo com a tabela 2.1 e no *output* apenas deve ser mostrada uma linha que ajude a identificar claramente o problema. Se o *output* ultrapassar uma linha, as linhas a mais serão ignoradas.

- Os serviços e *hosts* - conjunto de entidades a monitorizar. Não comunicam diretamente com a lógica sem recorrer a *plugins*. Os procedimentos executados pelos *plugins* têm sempre estas entidades como alvo de monitorização.

2.3.1.3 Funcionamento

Nesta secção pretende-se apresentar o modo de funcionamento do Nagios, salientando os ficheiros e diretórios mais importantes. Na figura 2.16 estão presentes os componentes essenciais na estrutura do Nagios.

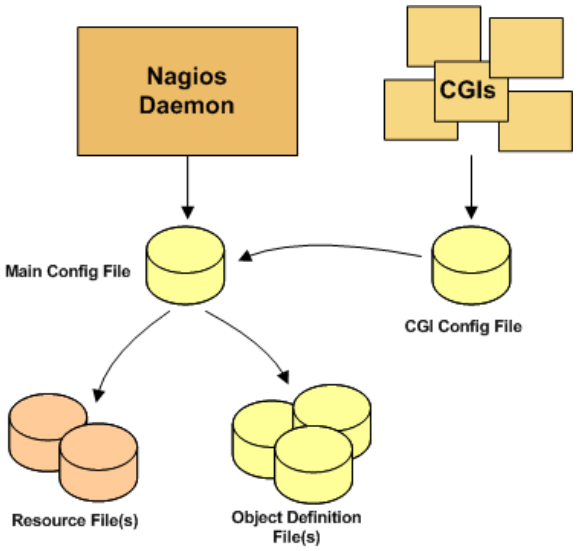


Figura 2.16: Funcionamento do Nagios [9]

O Nagios recorre a um *daemon* para atualizar periodicamente as informações que vai fornecendo. Um *daemon* é um processo do sistema operativo que é executado em *background*. Este *daemon* acede ao ficheiro central de configurações (/etc/nagios/nagios.cfg) para obter informações acerca do modo de operar, bem como acerca da localização dos outros ficheiros relevantes. Os ficheiros CGI são responsáveis pela gestão da interface *web* e, também, obtêm outras informações necessárias, recorrendo ao ficheiro central de configurações.

Existem mais dois componentes importantes na estrutura de funcionamento do Nagios - os ficheiros de recursos e os ficheiros de definição de objetos. Os primeiros são utilizados para guardar informação sensível pertencente aos utilizadores, os segundos definem os objetos. Estes ficheiros de objetos identificam o que é monitorizável e como o fazer. São entendidos pelo Nagios como as entidades envolvidas no processo de monitorização, ou seja, podem ser *hosts*, serviços, grupos de *hosts*, contactos, grupos de contactos, *scripts* ou qualquer outra entidade interveniente no processo.

Nas parágrafos anteriores falou-se de *plugins* e de servidores de monitorização mas não foram referidos os procedimentos para executar os *plugins*. Os responsáveis pela execução dos *plugins* são os agentes de monitorização, cujos mais comuns são o Nagios Remote Plugin Executor (NRPE) e o NSClient++.

2.3.1.4 NRPE

O NRPE [10] é o agente de monitorização para *hosts* cujo sistema operativo seja Linux. O modo de funcionamento está sucintamente descrito na figura 2.17. Em suma, o servidor de monitorização faz referência ao *plugin* check_nrpe e este estabelece uma ligação segura sobre o túnel SSL [38] com o agente de monitorização. Posteriormente, o agente de monitorização executa os *plugins* corretos e devolve o resultado de novo ao NRPE, o qual o reencaminha para o *plugin*

check_nrpe. O agente pode encontrar-se na mesma máquina que o servidor ou numa máquina acessível remotamente.

Para que o processo descrito possa realizar-se é necessário instalar no servidor de monitorização o *plugin* check_nrpe e no terminal que se pretende monitorizar o módulo NRPE.

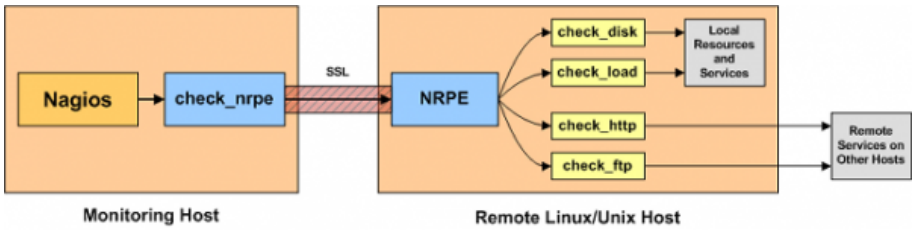


Figura 2.17: Modo de funcionamento do NRPE [10]

2.3.1.5 NSClient++

O NSClient++[51] é uma ferramenta em tudo semelhante ao NRPE, destinada a *hosts* cujo sistema operativo seja o Windows. O modo de funcionamento é idêntico mas, devido às alterações necessárias para a adaptação ao sistema Windows, ao *plugin* do servidor de monitorização dá-se o nome de check_nt.

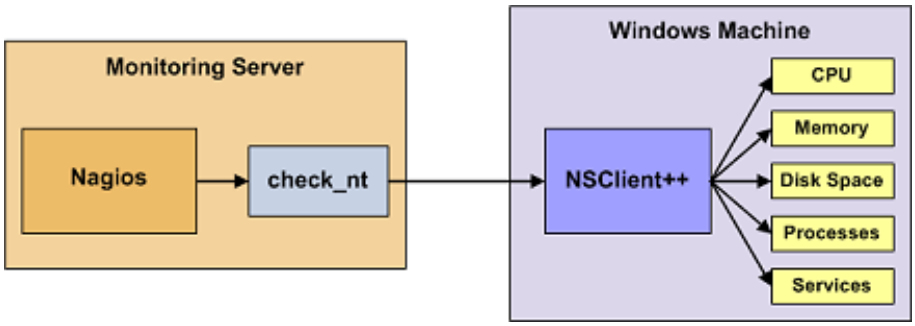


Figura 2.18: Modo de funcionamento do NSClient++ [11]

2.3.2 Desenvolvimento de um módulo de raiz

Outra solução possível prende-se com o desenvolvimento de um módulo IPBrick que não recorra a aplicações de terceiros. Esta solução implica o desenvolvimento de um novo módulo IPBrick com as duas componentes referidas na secção 2.2.4 - a interface *web* e a base de dados.

2.3.2.1 Arquitetura

Para além das duas entidades que estão presentes em todos os módulos IPBrick, este módulo em particular implicaria a intervenção de um conjunto de ficheiros responsável pela obtenção de

valores do sistema - os *scripts*. Assim, o módulo a desenvolver teria a arquitetura semelhante à representada na figura 2.19.

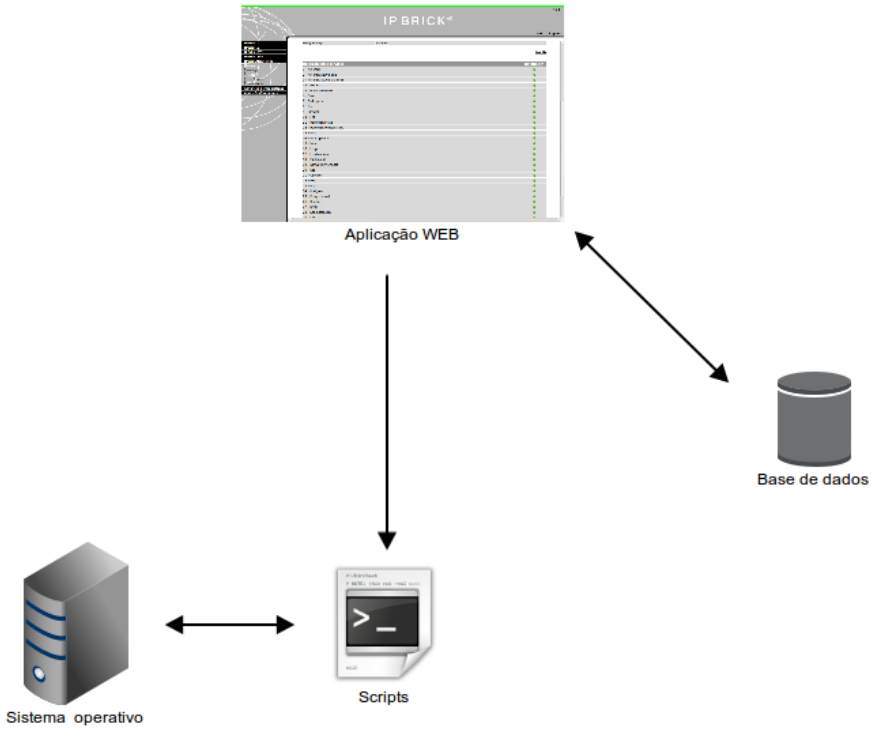


Figura 2.19: Arquitetura de um módulo IPBrick

Assim, as entidades que integrariam o módulo desempenhariam as seguintes funções:

- Aplicação *web* - permitiria a interação do utilizador com o módulo e o sistema. Numa primeira fase apresentaria uma lista com os serviços a serem analisados, possíveis configurações de parâmetros a ter em conta nas verificações aos serviços e uma opção que despontaria o início da análise ao sistema. Numa segunda fase, surgiriam os resultados da análise aos serviços seleccionados previamente;

- Base de dados - agregaria os dados de configuração e as informações acerca dos *scripts* a executar. A geração da interface *web* é dinâmica e baseada na informação contida na base de dados;

- *Scripts* - efetuariam a análise aos serviços e ao sistema. Os *scripts* estão para esta solução, como os *plugins* estão para o Nagios. Cada *script* executaria a verificação de um parâmetro do sistema ou de um serviço em particular através de chamadas ao sistema operativo;

- Sistema operativo - seria através das chamadas ao sistema operativo que os *scripts* efetuariam as verificações. Como já referido, a IPBrick tem como base o sistema operativo Linux

Debian, e,por isso, os comandos Unix seriam utilizados para a obtenção da informação necessária à análise.

2.3.2.2 Funcionamento

Já introduzidas as entidades que formariam o módulo, importa salientar como se realizaria o processo de verificação do sistema e dos serviços.

O processo de análise iniciar-se-ia com a disponibilização da informação disponível na base de dados ao utilizador através da interface *web*. Esta informação resultaria numa lista de serviços, de entre os quais o utilizador escolheria os que desejaria ver analisados. Após a submissão da escolha do utilizador, seriam invocados, um a um, os *scripts* correspondentes. Por fim, os valores obtidos nas verificações executadas pelos *scripts* seriam mostrados ao utilizador, organizados em forma de relatório.

2.3.2.3 Solução recorrendo ao Simple Network Management Protocol (SNMP)

Mesmo recorrendo a uma solução que implica o desenvolvimento do módulo de raiz, poder-se-ia recorrer a um protocolo de monitorização bem conhecido para efetuar a verificação ao sistema e aos serviços - o SNMP.

A utilização do protocolo SNMP é mais adequada à monitorização contínua do servidor. Como esta monitorização contínua não é um dos requisitos indispensáveis, a esta solução é dado menos ênfase.

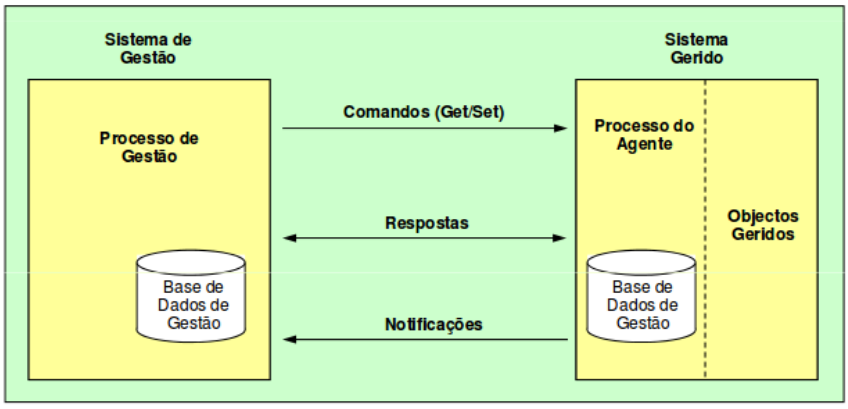


Figura 2.20: Arquitetura SNMP [12]

A arquitetura é semelhante à do Nagios, já apresentado na secção 2.3.1. Neste caso, o cliente é denominado agente (*Agent*) e o servidor é denominado gestor (*Manager*). O gestor pode efetuar pedidos ao agente. O agente responde aos pedidos e pode enviar alertas (*trap*) para valores alterados.

A estrutura das entidades monitorizadas é definida na Management Information Base (MIB) [52] que está presente no agente. Uma MIB é um conjunto de informações organizadas hierarquicamente. A cada parâmetro analisado dá-se o nome de objeto e cada objeto contém um identificador único denominado Object ID (OID).

Existem múltiplas MIB disponibilizadas para monitorizar os objetos mais comuns, mas existe também a possibilidade de desenvolver uma MIB para monitorizar um objeto específico.

2.4 Comparação das soluções

Nesta secção pretende-se analisar as vantagens e desvantagens das soluções apresentadas nas secções anteriores. Esta análise será essencialmente focada nos seis pontos seguintes.

2.4.1 Linguagem

Um dos parâmetros a ter em conta ao optar por desenvolver um módulo de raiz é a linguagem a utilizar. A escolha da linguagem tem uma influência relevante porque dela dependem os tempos de execução dos *scripts* e a complexidade do código, entre outros fatores.

Neste campo, o Nagios apresenta uma vantagem porque, tal como referido na secção 2.3.1, permite a utilização de qualquer linguagem de programação, compilada ou não, ao invés do módulo desenvolvido de raiz, cujos *scripts* têm de ser apresentados na mesma linguagem. Este aspeto confere um grau de versatilidade mais elevado no Nagios.

2.4.2 Reutilização de scripts disponíveis

Outro parâmetro importante que pode reduzir a quantidade de trabalho, e consequentemente o tempo de desenvolvimento, é a reutilização de *scripts* disponíveis.

As comunidades de desenvolvimento de *plugins* para o Nagios divulgam uma grande quantidade de *scripts* que podem ser reutilizados. Mas uma vez tomada a opção pelo desenvolvimento de um módulo de raiz, apenas os *plugins* desenvolvidos na linguagem escolhida podem ser reaproveitados.

2.4.3 Apoio ao desenvolvimento

No desenvolvimento dos *scripts* podem surgir algumas dúvidas relacionados com a metodologia ou a linguagem.O desenvolvimento de *plugins* para o Nagios apoia-se em várias comunidades de programadores que contêm resoluções dos problemas mais comuns, versões anteriores de produtos e em pessoas mais experientes.

2.4.4 Licenciamento

Um factor importante para um produto como o sistema operativo IPBrick é o do licenciamento relacionado com a integração de uma ferramenta *third-party*. Optando-se pela solução que envolve

o desenvolvimento do módulo de raiz, este problema não existe. Mas, se a opção tomada for a utilização do Nagios, então, o licenciamento e as condições a que o produto fica sujeito são factores relevantes a ter em conta.

As licenças do software a usar devem enquadrar-se na estratégia da empresa. Neste momento, o Nagios Core encontra-se protegido pela licença GNU General Public License (GNU GPL) [53] que permite utilizar e alterar livremente a ferramenta, mas todo o software desenvolvido a partir dele tem de ser distribuído com a mesma licença. Um dos princípios do software presente nos servidores IPBrick é tornar impossível a visualização do código desenvolvido pela empresa para evitar cópias e para garantir toda a segurança no acesso à informações dos utilizadores.

2.4.5 Agentes de monitorização

A utilização de agentes de monitorização é obrigatória no Nagios, o que implica a instalação de *software* adequado no terminal monitorizado. Já, uma solução desenvolvida de raiz não implica a instalação de qualquer *software* extra que não seja o módulo.

2.4.6 Adaptação às necessidades do produto

De um ponto de vista funcional, a utilização de uma ferramenta já produzida reduz o grau de liberdade para a adaptação às necessidades do produto. Já o desenvolvimento de um módulo de raiz permite atingir todos os aqueles requisitos com maior eficácia e segurança.

2.4.7 Resumo das vantagens e desvantagens

Após uma análise frente a frente dos seis pontos comparativos, apresenta-se um resumo das vantagens e desvantagens de cada uma das soluções acompanhado de uma tabela de comparação.

2.4.7.1 Vantagens da utilização do Nagios

O Nagios é uma ferramenta muito utilizada no plano de monitorização de redes e servidores. Em comparação com outras ferramentas semelhantes tem a vantagem de, na sua estrutura, estar prevista a utilização de *plugins*. Estes *plugins*, que podem até ser desenvolvidos por pessoas com o mínimo de conhecimento na área, dão um elevado grau de liberdade ao utilizador. Esta facilidade levou ao surgimento de diversas comunidades de programadores interessadas em desenvolver *plugins* para a deteção de falhas nos serviços mais utilizados. Assim, o elevado número de *plugins* já existentes é um valor que pode diminuir a carga de trabalho deste projeto.

A arquitetura do Nagios também concede uma vantagem ao nível da flexibilidade de desenvolvimento, isto é, os *plugins* podem ser desenvolvidos em qualquer linguagem, compilada ou não, e isso dá liberdade ao programador para utilizar a melhor linguagem, ou até desenvolver diferentes *plugins* com diferentes linguagens se achar necessário. Outra bonificação proveniente da utilização em larga escala é a facilidade de obter documentação e apoio técnico da comunidade de programadores.

2.4.7.2 Desvantagens da utilização do Nagios

Recorrer ao Nagios para solucionar a deteção de falhas também acarreta desvantagens que devem ser tidas em conta. A principal desvantagem é a integração de um software exterior à empresa. Este fator pode ter consequências mais gravosas se se der importância à situação das licenças de utilização, que, num produto como a IPBrick, é um fator demasiado importante para ser descartado. Outra desvantagem é a obrigatoriedade de instalar em cada servidor IPBrick um dos agentes de monitorização apresentados.

2.4.7.3 Vantagens da utilização do módulo desenvolvido de raiz

Uma das desvantagens referidas relativamente ao Nagios é a obrigatoriedade de inserção de *software* externo no código do sistema operativo IPBrick. Esta situação não ocorre com o módulo desenvolvido de raiz, inteiramente produzido ao longo deste projeto, evitando, entre outros, problemas referentes a licenças. Esta mesma solução, é possível adaptá-la às necessidades específicas do módulo, evitando funcionalidades não utilizadas a ocupar recursos do sistema.

2.4.7.4 Desvantagens da utilização do módulo desenvolvido de raiz

Relativamente às desvantagens de uma solução desenvolvida de raiz, a falta de flexibilidade quando comparada com o Nagios, que permite desenvolver os *plugins* em qualquer linguagem, é um factor a ter em conta. Nesta solução todos os *scripts* têm que ser desenvolvidos com a linguagem escolhida de início.

2.4.8 Opção tomada

Como as duas soluções - Nagios e o Desenvolvimento de um módulo de raiz - apresentam vantagens e desvantagens na sua utilização, é importante ao tomar a decisão escolher aquela que seja mais de acordo com o que a empresa e seus responsáveis achem mais relevante.

O Nagios tem evoluído muito devido à contribuição das comunidades de programadores, mas apresenta uma desvantagem associada à implementação de uma ferramenta sujeita a possíveis alterações na licença de utilização num sistema operativo baseado no conceito *open-source* e que não depende de licenças externas.

Por outro lado, a solução desenvolvida de raiz tem a vantagem de poder ser adaptada aos requisitos e às necessidades concretas da empresa. A tabela 2.2 contém um resumo da comparação entre as duas soluções.

Tabela 2.2: Comparação entre as duas soluções apresentadas - Nagios e o Desenvolvimento de um módulo de raiz

Nagios	**Desenvolvimento de um módulo de raiz**
Os plugins podem ser desenvolvidos em qualquer linguagem, compilada ou não	Todos os scripts devem ser desenvolvidos na mesma linguagem, que deve ser escolhida na fase pré-desenvolvimento
É possível a reutilização de vários plugins disponíveis no site oficial do Nagios	É necessário desenvolver todos os scripts ou, possivelmente, reutilizar alguns plugins desenvolvidos na linguagem escolhida
Apoio da comunidade de programação para Nagios na resolução de dúvidas	Uma vez que o desenvolvimento é realizado de raiz não existe apoio de nenhuma comunidade dedicada exclusivamente a este fim nem versões anteriores, mas existem situações semelhantes relatadas em comunidades de apoio à programação
Sujeito a licenciamento GNU GPL	Não está sujeito a nenhuma restrição de licenciamento
Exige a instalação de agentes de monitorização	A instalação de agentes de monitorização não é obrigatória
O facto de ser uma solução já desenvolvida não permite a existência de liberdade para adaptação às necessidades específicas do produto	O facto de ser uma solução desenvolvida de raiz permite a adaptação às necessidades específicas do produto

Ponderados os prós e contras, optou-se por desenvolver um módulo de raiz porque as desvantagens apresentadas pelo Nagios assumem tais restrições que esta opção teve de ser descartada. O desenvolvimento de uma solução de raiz evita os problemas com licenças e também permite que o módulo possa ser melhorado quando necessário, acompanhando a evolução do *software* desenvolvido pela empresa. As maiores capacidades de adaptação e de modulação às necessidades específicas do produto são os outros fatores que levam a esta escolha.

Capítulo 3

Módulo de diagnóstico e despiste de problemas

Nos capítulos anteriores foi descrito o problema e foram apresentadas as ferramentas que poderiam ser as suas soluções. Este capítulo foca o trabalho prático desenvolvido ao longo do semestre, destacando alguns conceitos essenciais e as diversas etapas ultrapassadas.

3.1 Planeamento

3.1.1 Metodologia de desenvolvimento

Para o desenvolvimento da solução ser mais coerente e com o menor número de contratempos possível, para além dos requisitos definidos, optou-se pela elaboração de uma lista de serviços presentes nos servidores IPBrick e pela atribuição de um valor de 0 a 10 correspondente à sua criticidade no sistema, sendo 0 um serviço de importância diminuta e 10 um de importância extrema.

A tabela 3.1 contém estes valores de criticidade para cada serviço, acordados junto com os responsáveis da empresa.

Tabela 3.1: Criticidade atribuída a cada serviço presente no sistema operativo IPBrick

Nome do serviço	Descrição	Criticidade [0-10]
Memória RAM, CPU e Disco rígido	Análise dos recursos disponíveis e sua opcupação	10
PostgreSQL	Verificar o acesso à base de dados	10
DNS	Servidor de resolução de nomes fundamental para a solução fornecida	10
LDAP	Servidor de gestão de utilizadores	10
Firewall	Análise às portas bloqueadas e portas permitidas	9
Asterisk + Kamailio	Dois serviços responsáveis pela comunicação VOIP	9
Qmail	Servidor de email	9
Ejabberd	Servidor de IM	9
Apache	Servidor web	9
hylaFAX + SMS	Serviços de comunicação por fax e sms	9
openVPN	Servidor VPN	9
Squid + SquidGuard + DansGuardian	Servidor Proxy	8
SpamAssassin + ClamaV	Verificador de spam e Antivirus	8
Virtualização	Servidor de virtualização	8
PrintServer	Servidor de impressão	8
SSH	Servidor de ssh	7
Crontab	Agendador de tarefas do linux	6
Samba + FTP	Servidor de partilhas de áreas de trabalho remoto e de ficheiros	6
SNMP	Serviço de monitorização remota	5
NTP	Servidor horário	4
Fetchmail	Cliente de email responsável por obter e-mails do ISP	4
Radius	Servidor de autenticação	3

Mesmo não optando pelo Nagios como ferramenta de Monitorização, achou-se importante o formato dos *scripts* a desenvolver ser semelhante ao dos *plugins* Nagios devido às vantagens que, mesmo que não sejam importantes para o sistema neste momento, podem vir a sê-lo no futuro. Assim, neste sistema, futuramente, se for necessário utilizar o Nagios como ferramenta de monitorização, os *scripts* podem ser reutilizados.

Como já foi descrito na secção 2.3.1, os *plugins* são *scripts* que verificam o estado dos serviços sob duas regras - apenas é apresentada uma linha de informação para o utilizador e o valor retornado após a execução deve ser de acordo com a tabela 2.1

3.1.2 Linguagem de programação utilizada

Relativamente ao desenvolvimento dos *scripts*, numa primeira fase, optou-se pela linguagem C/C++, mas logo se percebeu que não se justificava a utilização de uma linguagem de baixo nível. Optou-se, então, por testar outras possibilidades como o Perl, o Python e o Hypertext PreProcessor (PHP), cujos ficheiros de execução não são compilados. A execução de qualquer uma destas linguagens é um pouco mais lenta comparativamente com os ficheiros compilados, mas chegou-se à conclusão que esse não é um fator demasiado relevante, até porque uma grande fatia do código fonte presente no sistema operativo IPBrick é desenvolvido em PHP. O ganho de eficiência que poderia surgir da utilização de uma linguagem de baixo nível perder-se-ia devido à existência de atrasos em outros *scripts* do sistema operativo.

Já, uma linguagem não compilada implica uma desvantagem que é a da privacidade do código não ser garantida, isto é, o código ficar exposto. Para este efeito, na IPBrick, recorre-se à ferramenta ZendGuard [54] que permite esconder o código PHP, dando a este ficheiro um forma semelhante ao ficheiro compilado. Desta maneira, o código encontra-se escondido mas ao mesmo tempo é possível executá-lo como se de um *script* PHP normal se tratasse. Foi este fator que levou à preferência do PHP como linguagem para o desenvolvimento dos *scripts* em detrimento das restantes linguagens apresentadas.

3.1.3 Estrutura dos ficheiros

Para manter a organização do código, definiu-se, previamente, a organização de diretórios que está representada na figura 3.1.

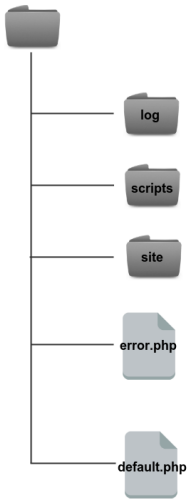


Figura 3.1: Estrutura de ficheiros os módulo a desenvolver

Os ficheiros referentes ao módulo desenvolvido são colocados num diretório que apenas diz respeito a este módulo - representado na figura pelo diretório de topo. No interior deste, existem dois ficheiros e mais três diretórios. Os dois ficheiros são o *default.php*, que contém as variáveis

e as configurações comuns a todos os *scripts*, e o *error.php*, que efetua uma conversão entre os códigos retornados pelos *scripts* e as mensagens mostradas ao utilizador. Os três diretórios são *log*, *scripts* e *site*. O primeiro contém o ficheiro de registo de ocorrências do módulo. O segundo contém todos os *scripts* desenvolvidos. Estes estão organizados de acordo com o serviço a que pertencem, ou seja, existe um diretório por cada serviço, e, no interior do diretório, encontram-se todos os *scripts* referentes a esse serviço. O terceiro contém as páginas *web* que constituem a interface *web* do módulo.

3.1.4 Execução dos scripts

Apresentada a linguagem a utilizar e a organização estrutural dos *scripts* desenvolvidos, mostra-se, agora, como podem ser executados. Como foi referido, optou-se pelo desenvolvimento dos *scripts* respeitando as regras dos *plugins* Nagios, para que, podendo ser útil no futuro, seja possível utilizar o Nagios como ferramenta de monitorização. Para maior flexibilidade, todos os *scripts* desenvolvidos aceitam argumentos de entrada que parametrizam o seu comportamento durante a execução.

Os argumentos de entrada aceites pelos *scripts* são apresentados na tabela 3.2.

Tabela 3.2: Argumentos de entrada aceites pelos *scripts*

Opção	Descrição
-N	Quando inserida esta opção o *script* devolve uma mensagem no formato de *plugin* do Nagios - escreve uma linha no output e retorna um valor *OK WARNING* ou *CRITICAL*
-w	Esta opção é seguida de um intervalo no formato que será apresentado de seguida. Se o valor obtido pertencer ao intervalo de valores, o retorno é o estado *WARNING*
-c	Esta opção é seguida de um intervalo no formato que será apresentado de seguida. Se o valor obtido pertencer ao intervalo de valores, o retorno é o estado *CRITICAL*

A definição de intervalos segue as normas utilizadas nos *plugins* Nagios. Na tabela 3.3 é apresentada a nomenclatura aceite na definição dos intervalos, acompanhada de uma breve descrição.

Tabela 3.3: Definição dos intervalos nos *scripts*

Intervalo	Gera alerta se k
x	$k \in \ ]-\infty, 0[\ \vee \ ]x, +\infty]$
x:	$k \in \ ]-\infty, x[$
~:x	$k \in \ ]x, +\infty]$
x:y	$k \in \ ]-\infty, x[\ \vee \ ]y, +\infty]$
@x:y	$k \in \ [x, y]$

Portanto, os parâmetros de saída de um *script* variam em função dos parâmetros de entrada. Apresentam-se dois exemplos de utilização:

E.g. php script1.php -N -w @50:90 -c @90:100

Este comando executará o *script1.php* com o *output* formatado para a utilização do Nagios. Se o valor lido se encontrar no intervalo [50,90] será retornado o estado *WARNING*, se o valor lido se encontrar no intervalo [90,100] será retornado o estado *CRITICAL*

E.g. php script2.php -w @50:90 -c @90:100

Este comando executará o *script2.php* com o *output* para o módulo desenvolvido, ou seja, não haverá *output* para o terminal, sendo o resultado copiado para um vetor que será lido pelo módulo.

3.1.5 Interface *web*

No momento anterior ao desenvolvimento da interface *web* associada ao módulo, elaborou-se um esboço que serviu de guia para evitar eventuais contradições ou incoerências com a restante interface *web* do sistema operativo IPBrick. Nesta secção apresenta-se esse esboço acompanhado da explicação das opções disponíveis.

Primeiro apresenta-se a interface *web* do módulo em geral, enquadrando-se na interface IPBrick, depois descrevem-se, com mais pormenor, as opções presentes em cada serviço.

3.1.5.1 Geral

Como se pode observar na figura 3.2, a interface *web* IPBrick disponibiliza as opções num menu localizado no lado esquerdo do ecrã. Este menu contem todas as opções divididas de acordo

com o módulo em que se encontram inseridas. Ficou acordado com os responsáveis da IPBrick SA que o módulo de diagnóstico e despiste de problemas seria inserido nas opções avançadas, na opção *System* > *Monitoring* > *Troubleshooting* .

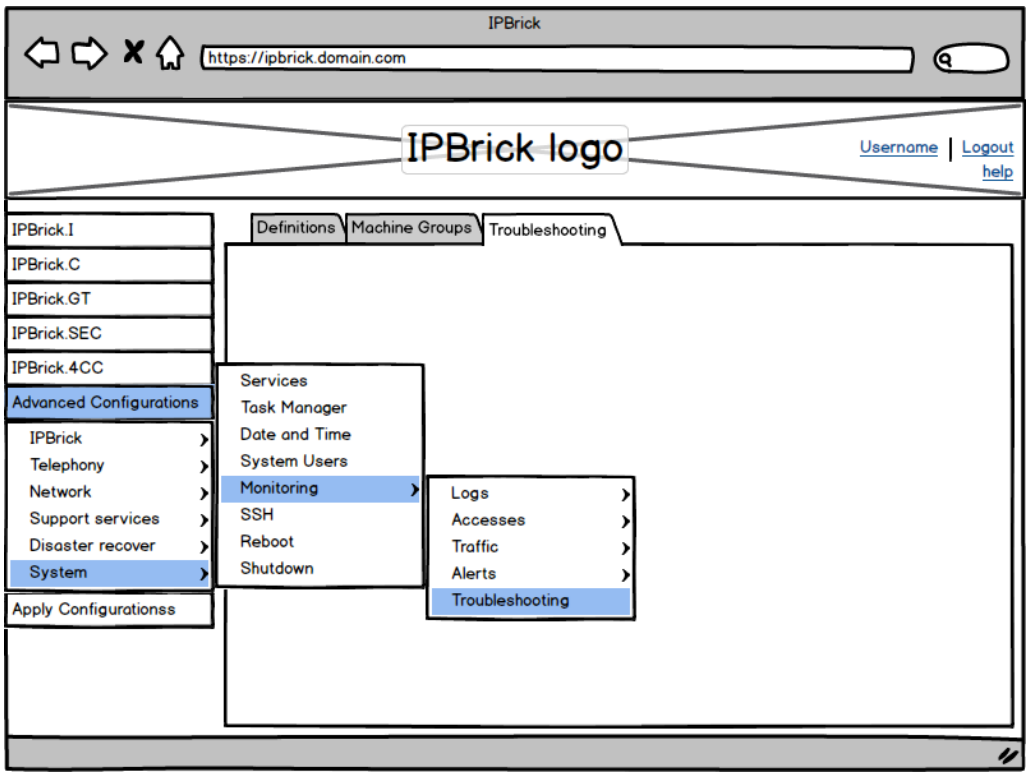


Figura 3.2: Menu IPBrick

Após o utilizador escolher esta opção será redirecionado para a página de configurações do módulo - figura 3.3. Neste ecrã, é apresentada a informação relativa aos serviços sobre os quais é possível realizar diagnóstico. A opção *Expand* permite obter informação detalhada sobre as verificações que podem ser executadas sobre um serviço. Quando esta opção é seleccionada, é exibido um ecrã semelhante à figura 3.4.

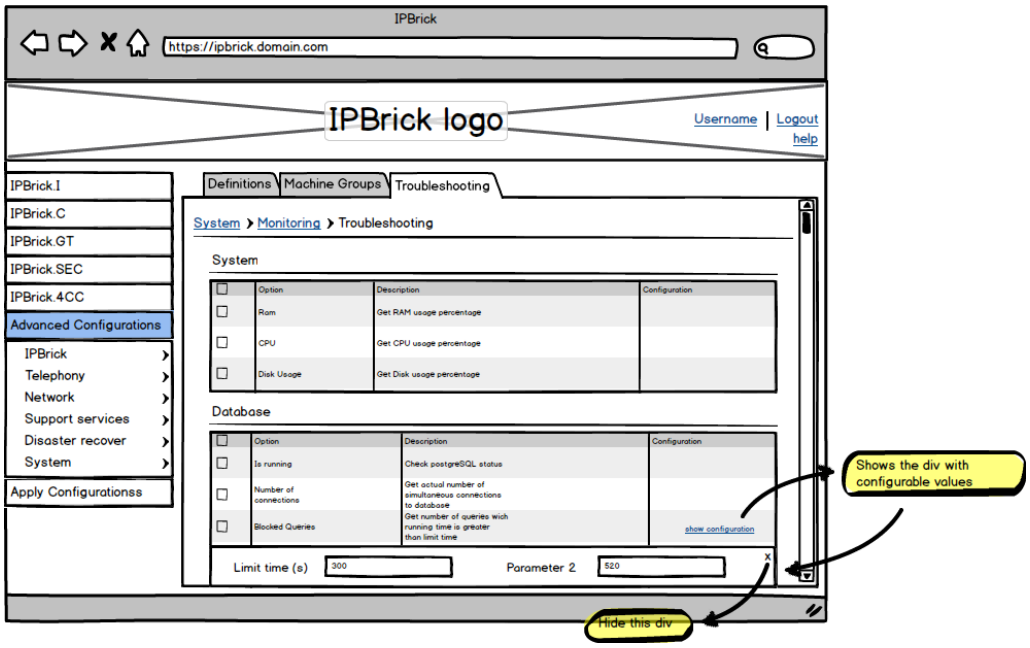


Figura 3.3: Página de apresentação dos serviços

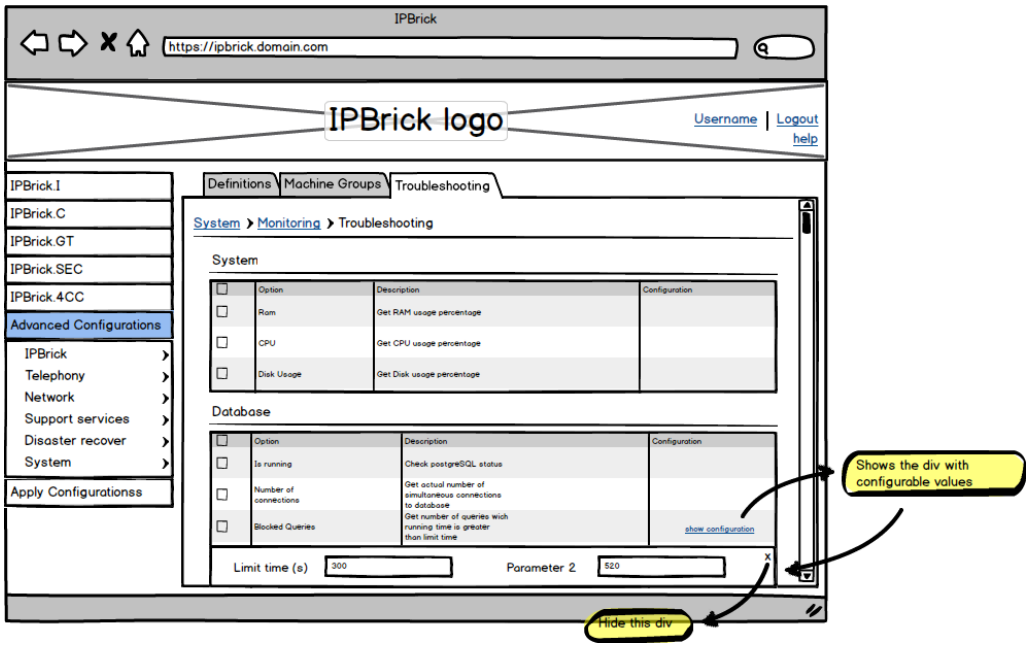


Figura 3.4: Detalhes de cada serviço

Em alguns parâmetros é possível definir valores de configuração através da opção *show configuration*. Quando é seleccionada esta opção, surge uma divisão no ecrã que permite alterar o valor de alguns parâmetros. A título de exemplo, na figura 3.5), é permitido alterar o valor dos parâmetros *Time Limit* e *Parameter 2*.

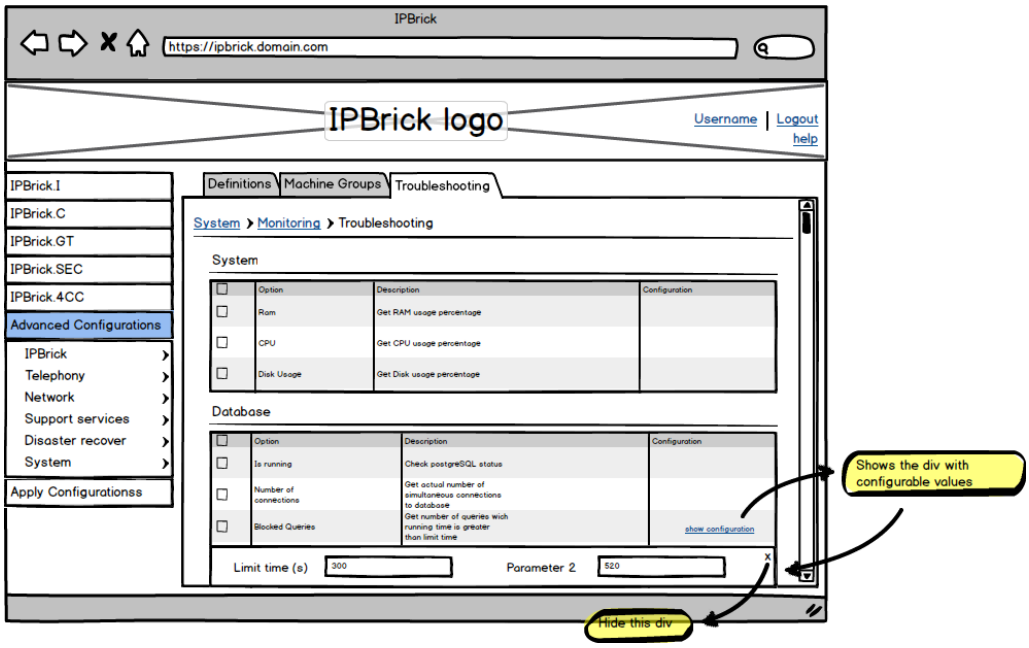


Figura 3.5: Parâmetros de configuração

Por fim, depois de seleccionados os serviços pretendidos para analise e as configurações necessárias, obtém-se um relatório, semelhante ao da figura 3.6, contemplando todas as informações recolhidas.

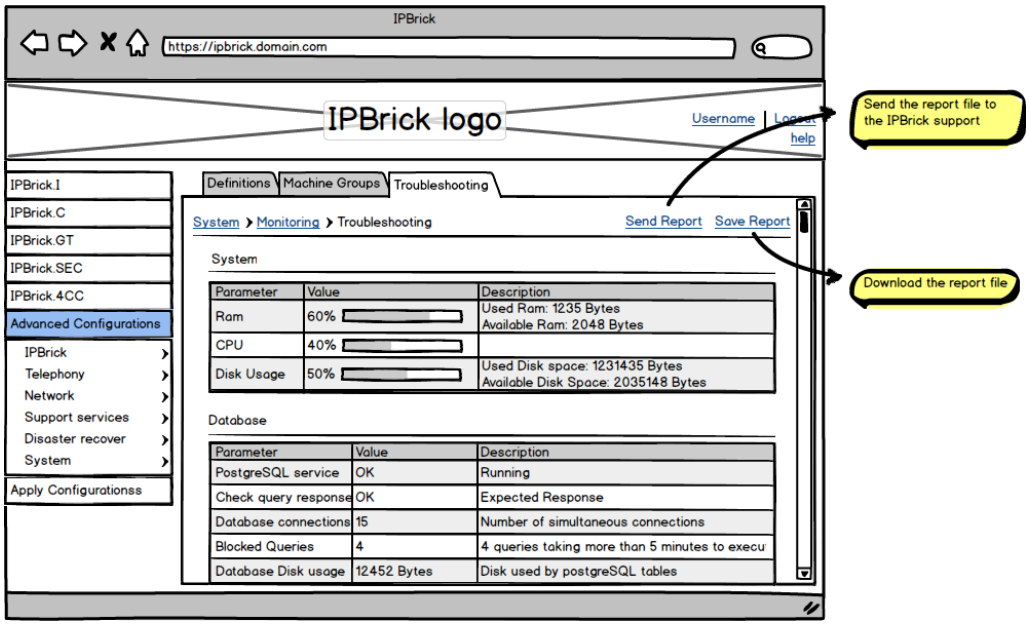


Figura 3.6: Excerto do relatório relativo aos parâmetros do correio eletrónico

O relatório obtido pode ser exportado para o formato *pdf* ou diretamente enviado através do correio eletrónico para o apoio tecnico da IPBrick. Estas duas opções encontram-se no lado esquerdo do menu superior.

Seguidamente analisa-se com mais detalhe as opções existentes para cada serviço.

3.1.5.2 Sistema

Essencialmente, os *scripts* desenvolvidos para a análise do sistema, visam obter dados relativos à utilização e disponibilidade dos recursos. Desenvolveram-se três *scripts* cuja função é obter os valores da percentagem ocupada de disco rígido, percentagem ocupada de processamento e percentagem ocupada de memória RAM. Estes valores são obtidos recorrendo às ferramentas disponíveis no sistema UNIX.

System Collapse

	Option	Description	Configuration
☑	RAM	Get ram memory usage	
☐	CPU	Get CPU usage	
☑	Hard disk	Get hard disk usage	

Figura 3.7: Área de selecção de parâmetros relativos às verificações do sistema

Com estes parâmetros é pretendido que a interface *web* seja semelhante à figura 3.7 no momento de configuração e à figura 3.8 no momento de visualização de informações recolhidas.

System

Parameter	Value	Description
Ram	60%	Used Ram: 1235 Bytes Available Ram: 2048 Bytes
CPU	40%	
Disk Usage	50%	Used Disk space: 1231435 Bytes Available Disk Space: 2035148 Bytes

Figura 3.8: Excerto do relatório relativo aos parâmetros do sistema

3.1.5.3 Base de dados

Pelos motivos já explicados, a base de dados é o serviço de que o sistema operativo IP-Brick mais depende. O próprio serviço PostgreSQL prevê o armazenamento de informações de estado em tabelas adequadas. A tabela que acumula as informações de estatísticas de utilização, bem como erros, comportamentos inesperados, registo de acessos e de *queries* tem o nome pg_stat_activity.

Os *scripts* desenvolvidos realizam consultas a esta tabela que, após processamento da informação recebida, geram os parâmetros seguintes:

- Estado do serviço (*Is running*) - permite obter o estado de execução do serviço PostgreSQL. O valor obtido tem o formato de *output* dos *plugins* Nagios - *OK*, *WARNING*, *CRITICAL* ou *UNKNOWN*;

- Verificação da resposta (*Check query response*) - executa uma *query* de teste que permite verificar se a resposta corresponde ao esperado; Este teste complementa o anterior uma vez que garante que o serviço se encontra em execução e também se encontra a responder aos pedidos;

- Número de ligações simultâneas (*number of connections*) - permite obter o número atual de ligações simultâneas à base de dados. O número de ligações à base de dados é um indicador que permite identificar tentativas de ataque ou acessos indevidos. e.g. A coexistência de 200 ligações à base de dados, numa rede em que apenas a solução IPBrick devia ter acesso a esta, pode ser um indicador de que a base de dados está a ser indevidamente acedida por outros;

- Número de *queries* bloqueadas (*blocked queries*) - permite obter o número de *queries* cujo tempo de execução é superior a um dado tempo limite. Este parâmetro permite identificar *queries* que se encontram em execução durante largos períodos de tempo. Uma *query* em execução durante largos períodos de tempo pode significar um bloqueio do sistema ou falta de capacidade de resposta e provoca um atraso nas *queries* que se encontram na posição seguinte da fila de espera;

- Número de *queries WAITING* (*waiting queries*) - permite obter o número de *queries* que se encontram no estado *WAITING*. O estado *WAITING* caracteriza um problema de concorrência que ocorre quando um *backend* se encontra bloqueado por outro causando atrasos na execução das *queries* da fila de espera;

- Espaço em disco (*hard disk space*) - permite obter o tamanho, em *Bytes*, ocupado pelas tabelas da base de dados. Uma vez que existe uma quantidade elevada de informações armazenadas na base de dados, é importante obter uma métrica do espaço em disco ocupado por estes dados para maior controlo do armazenamento necessário.

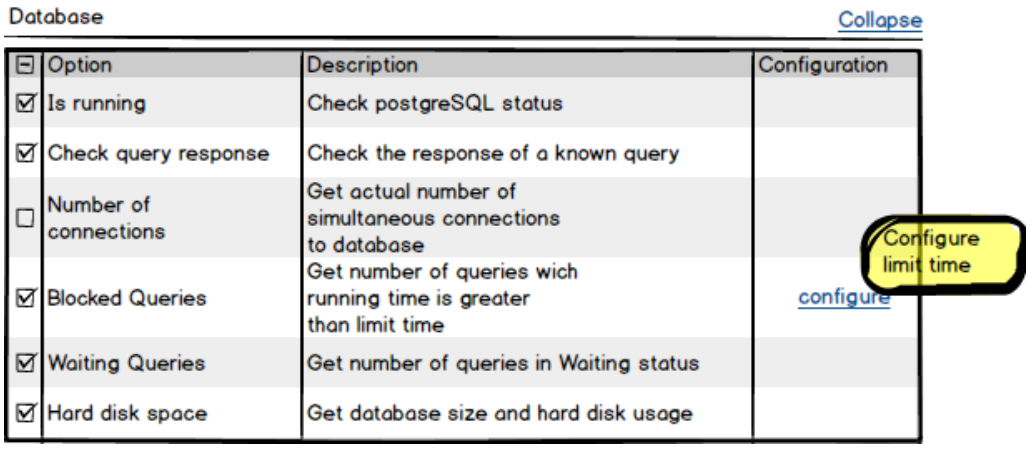


Figura 3.9: Área de introdução de parâmetros relativos à base de dados

Todos estes parâmetros devem ser mostrados numa interface de configuração semelhante à figura 3.9 e, posteriormente, deve surgir a interface com a informação recolhida, semelhante à figura 3.10

Database

Parameter	Value	Description
Is running	OK	postgreSQL service is running
Check query response	OK	postgreSQL service is responding as expected
Number of connections	13	There are 13 active connections
Blocked Queries	2	There are 2 queries which running time is greater than 5 minutes
Waiting Queries	1	There is 1 query with Waiting status
Hard disk space	13925 Bytes	The databases are using 13925 Bytes of hard disk space

Figura 3.10: Excerto do relatório relativo aos parâmetros da base de dados

3.1.5.4 DNS

Já foi descrita neste documento a importância do serviço DNS para a solução IPBrick. Seguidamente procede-se à descrição do planeamento, ao nível da interface *web*, relativo a este serviço. A interface *web* deve prever a análise aos seguintes pontos:

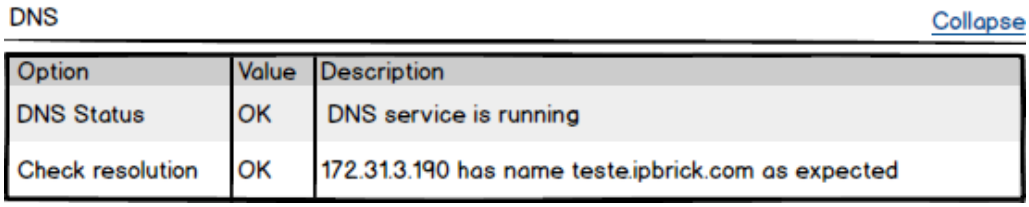
- Estado do serviço DNS (*DNS Status*) - indica o estado de execução do serviço bind9. O valor obtido tem o formato do *output* dos *plugins* Nagios - *OK*, *WARNING*, *CRITICAL* ou *UNKNOWN*;

- Resolução de nomes (*Check Resolution*) - permite verificar se o servidor de DNS se encontra em execução e a responder de acordo com o esperado. O teste é efetuado consultando a base de dados IPBrick para obter um nome na notação FQDN e o respetivo endereço IP. Seguidamente, realiza-se uma consulta ao servidor bind9 com o nome e verifica-se se a resposta do servidor é o endereço IP presente na base de dados. Se os endereços coincidirem, indica-se que a resolução foi executada com sucesso. Se a resposta não coincidir, indica-se que o servidor DNS não respondeu corretamente e, por isso, existe uma anomalia no serviço.

DNS Collapse

☐	Option	Description	Configuration
☑	DNS Status	Is DNS service running?	
☐	Check resolution	Name resolution is ok	

Figura 3.11: Área de selecção de parâmetros relativos às verificações do servidor de DNS


DNS Collapse

Option	Value	Description
DNS Status	OK	DNS service is running
Check resolution	OK	172.31.3.190 has name teste.ipbrick.com as expected

Figura 3.12: Excerto do relatório relativo aos parâmetros do servidor DNS

Os parâmetros devem ser mostrados numa interface de configuração semelhante à figura 3.11 e, posteriormente, deve surgir a interface com a informação recolhida, semelhante à figura 3.12.

3.1.5.5 DHCP

O planeamento da interface *web* - figuras 3.13 e 3.14- relativa ao DHCP prevê a obtenção de, apenas, um parâmetro - o estado de execução do serviço (*DHCP Status*). Uma vez mais, o valor obtido para este parâmetro tem o formato do *output* dos *plugins* Nagios - *OK*, *WARNING*, *CRITICAL* ou *UNKNOWN*. Não é possível realizar outros testes a este serviço porque isso implicaria a emissão de mensagens DHCP que poderiam interferir com outros dispositivos da rede.

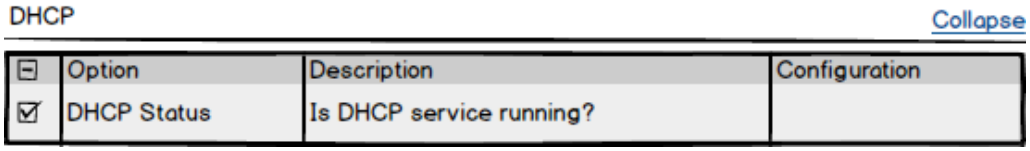


Figura 3.13: Área de selecção de parâmetros relativos às verificações do serviço DHCP

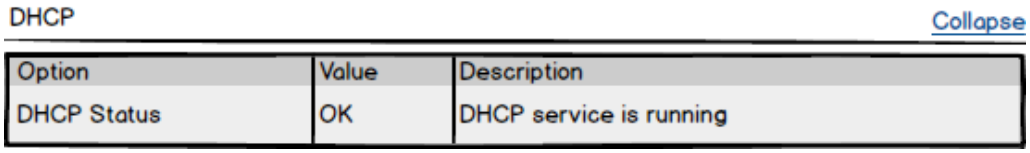


Figura 3.14: Excerto do relatório relativo aos parâmetros do serviço DHCP

3.1.5.6 LDAP

Relativamente ao serviço LDAP, os testes ao seu próprio estado de execução não são suficientes para garantir o funcionamento correto, já que é necessário garantir que os serviços de que este depende se encontram também em execução, nomeadamente o Automount. Assim, a verificação a realizar engloba o estado do serviço openLDAP, bem como o estado do serviço Automount. O valor obtido para esta verificação tem o formato do *output* dos *plugins* Nagios - *OK*, *WARNING*, *CRITICAL* ou *UNKNOWN*.

A interface *web* de configuração e de visualização deverão assemelhar-se às figuras 3.15 e 3.16, respetivamente.

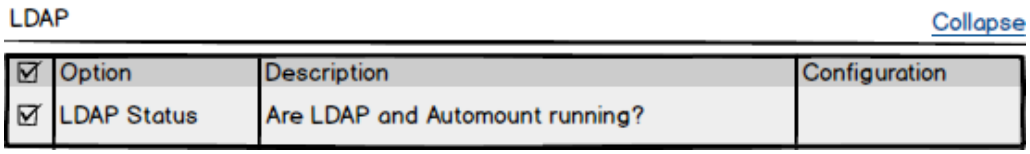


Figura 3.15: Área de selecção de parâmetros relativos às verificações do serviço LDAP

LDAP Collapse

Option	Value	Description
LDAP Status	OK	LDAP service and Automount are running

Figura 3.16: Excerto do relatório relativo aos parâmetros do serviço LDAP

3.1.5.7 Firewall

A opção que está prevista na interface *web* para a verificação dos parâmetros relativos à *firewall* é a indicação do estado das portas, ou seja, é mostrada uma listagem com as portas que se encontram abertas. Esta informação permite detetar as portas susceptíveis a ataques.

As interfaces *web* de configuração e de visualização dos resultados deverão assemelhar-se às figuras 3.17 e 3.18, respetivamente.

Firewall Collapse

☑	Option	Description	Configuration
☑	Open ports	Check which ports are open	

Figura 3.17: Área de selecção de parâmetros relativos às verificações da firewall

Firewall Collapse

Option	Value	Description
Open ports	Port 1000 open Port 2000 open Port 2001 open Port 5430 open	

Figura 3.18: Excerto do relatório relativo aos parâmetros da firewall

3.1.5.8 VoIP

No servidor de VoIP, ao invés de outros serviços nos quais a indicação de execução é suficiente para afirmar o bom funcionamento, é necessário obter diversos parâmetros. Os parâmetros seguintes estão previstos no planeamento da interface *web* :

- Estado do serviço (*Status*) - permite verificar o estado de execução dos serviços Asterisk e Kamailio. Uma vez mais, os valores obtidos têm o formato do *output* dos *plugins* Nagios;

- Estado das portas (*Ports state*) - permite obter o estado das portas utilizadas para a sinalização SIP e Session Initiation Protocol Secure (SIPs). Se ambas as portas se encontrarem bloqueadas pela *firewall*, trata-se de uma indicação clara de que o serviço de telefonia não se encontra em funcionamento;

- Telefones registados (*Registered phones*) - permite obter o número de telefones registados, bem como quantos destes se encontram *online*. Um elevado número de telefones registados pode ser um indicador de que existem registos não autorizados no sistema;
- Objetos (*Objects*) - permite obter uma lista de todos os objetos registados no serviço Asterisk. Desta forma, obtém-se uma perspectiva geral de todos os elementos registados no Asterisk, sejam eles agentes ou SIP *trunks*;
- Chamadas ativas (*Number of active calls*) - permite obter o número de chamadas ativas no instante da análise. Um número elevado de chamadas ativas sugere um possível ataque à solução. e.g. Numa organização com 50 telefones registados, o facto de existirem 200 chamadas ativas simultaneamente é um indicador claro de que algo não está correto;
- Verificação dos *Trunks* (*Trunk*) - permite obter o estado dos *Trunks* configurados no serviço Asterisk. É usado o *sipsak* para verificar a ligação aos servidores de destino. Muitos dos problemas relacionados com o Asterisk prendem-se com a incapacidade de ligação com os SIP *trunks*. Se a ligação com estas entidades não for possível, é expectável que não seja possível efetuar chamadas para os telefones registados no servidor de destino;
- Licença do *codec* G729(*G729 License*) - permite controlar a utilização da licença do *codec* G729 usado para o *audio*. A licença limita a sua utilização a um número restrito de chamadas simultâneas. Esta indicação permite identificar a percentagem de utilizadores deste *codec* relativa ao limite imposto pela licença;

VoIP Collapse

☑	Option	Description	Configuration
☑	Status	Are Asterisk and Kamailio services running?	
☑	Ports state	Check if SIP and SIPs ports are open	
☑	Registered phones	Get list of registered and online phones	
☑	Number of active calls	Get number of active calls	
☑	Trunks	Check trunks connectivity	
☑	G729 License	Check G729 license usage	
☑	Codecs	Check codecs in use	
☑	Agents	Get list of configured agents	
☑	Online Agents	Get list of online agents	
☑	Do not disturb	Get list of virtualhosts	
☑	Telephony cards	Get telephony cards status	

Figura 3.19: Área de introdução de parâmetros relativos ao serviço VoIP

- *Codecs* em uso (*Codecs*) - permite obter uma lista dos *codecs* em uso. Surge uma lista dos *codecs* que estão a ser usados nas chamadas que estão estabelecidas;
- Agentes configurados (*Agents*) - permite obter uma lista dos agentes configurados;
- Agentes *online* (*Online Agents*) - permite obter uma lista com os agentes configurados que se encontram *online*;
- Utilizadores com a opção *Do not disturb* ativa (*Do not disturb*) - permite obter a contagem dos utilizadores com a opção *Do Not Disturb* ativa. Se este número for elevado é indicador de que muitas extensões da rede se encontram incontactáveis;
- Placas de telefonia (*Telephony cards*) - permite obter o estado das placas de telefonia. É possível detetar as anomalias relacionadas com o *hardware*. Muitos problemas do serviço VoIP referem-se a falhas nas placas de telefonia.

A interface *web* deverá assemelhar-se às figuras 3.19 e 3.20.

VoIP Collapse

Option	Value	Description
Status	OK	Asterisk and Kamailio are runni
Ports state	SIP port is open SIPs port is open	
Registered phones	Number of registered phones: 10 Number of online phones: 5	
Number of active ca	5	
Trunks	Trunk sip.testvoip.com is OK	
G729 License	Licensed channels: 0	Encoders in use: 0 Decoders in use: 0
Codecs	Codecs used now: No codecs in use.	
Agents	No Agents are configured in agents.c	
Online Agents	No Agents are configured in agents.c	
Do not disturb	0 results found.	
Telephony cards	UP	

Figura 3.20: Excerto do relatório relativo aos parâmetros do serviço VoIP

3.1.5.9 Correio Electrónico

Para a obtenção de um relatório de informação relacionada com as funcionalidades de email é necessário englobar a análise dos serviços Courier-POP, Courier-POP-SSL, Courier-IMAP, Courier-IMAP-SSL e Qmail, pelos motivos explicados na secção anterior. Desta forma, seguindo o raciocínio aplicado em outros serviços, na interface de configuração da execução do módulo deverão surgir as opções representadas na figura 3.21.

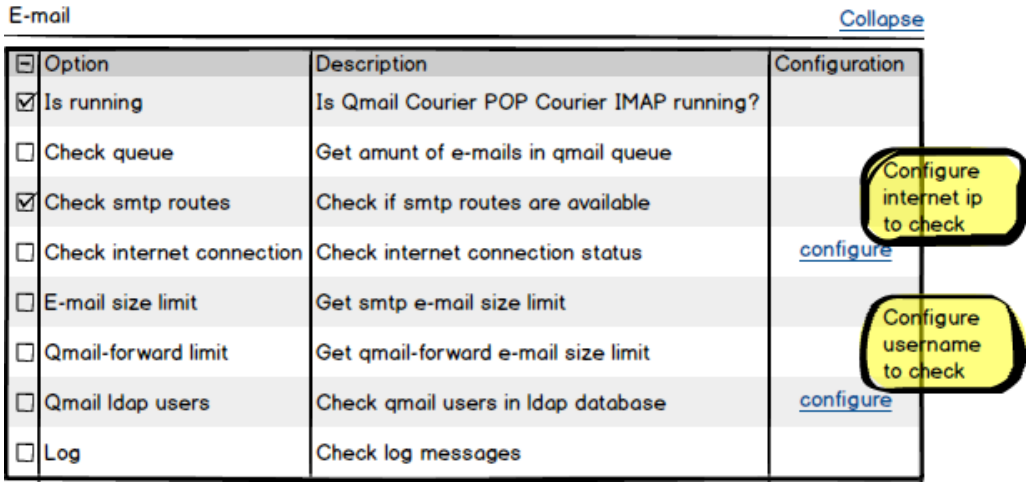


Figura 3.21: Área de introdução de parâmetros relativos ao serviço de correio eletrónico

Estas configurações permitem seleccionar qual ou quais as informações que devem ser obtidas. As configurações disponíveis são:

- Estado do serviço (*Is running*) - obtém o estado de execução dos serviços relacionados com o email, Qmail, Courier-POP, Courier-IMAP, Courier-POP-SSL e Courier-IMAP-SSL;
- Verificação da fila de espera (*Check queue*) - obtém o número de mensagens de correio eletrónico que se encontram na fila do Qmail;
- Verificação das rotas SMTP(*Check SMTP routes*) - Testa a conetividade com o destino das rotas SMTP. Quando um e-mail é enviado, o MTA consulta um servidor DNS para obter o IP do MTA de destino. Se esse IP não estiver disponível nos servidores DNS consultados pelo remetente, terá que ser configurada uma rota SMTP para que as mensagens continuem a ser entregues. Esta opção verifica a conetividade com o IP de destino das rotas SMTP executando uma ligação *telnet* para a porta default do SMTP, a porta 25;
- Verificação da ligação à Internet(*Check internet connection*) - Testa a conetividade com um endereço exterior à rede. Por vezes, não é possível enviar emails para o exterior devido a uma falha na ligação à Internet. Esta opção permite testar a acessibilidade a um determinado endereço IP através do comando *ping*. O endereço IP de destino pode ser alterado no botão "configure";
- Tamanho limite de *email* enviado (*E-mail size limit*) - Obtém o valor configurado como limite de envio. As mensagens de correio eletrónico cujo tamanho supere este valor são rejeitadas pelo Qmail;
- Tamanho limite de *email* reencaminhado (*Qmail-forward limit*) - Obtém o valor configurado como limite de reencaminhamento. As mensagens de correio eletrónico cujo tamanho supere este valor são rejeitadas pelo Qmail;

- Vericação do utilizador LDAP(*Qmail LDAP users*) - Verifica os parâmetros Qmail no LDAP para um determinado utilizador. Este utilizador pode ser alterado seleccionando a opção "configure";

- Estado das mensagens (*Log*) - Verifica as mensagens que surgem no log que permitem identificar, por exemplo, as razões de insucesso no envio de um email.

Depois das configurações submetidas, o relatório gerado será semelhante ao da figura 3.22.

As opções que indicam estados são apresentadas com os quatro estados possíveis do Nagios, *OK*, *WARNING*, *CRITICAL* e *UNKNOWN*. As restantes opções são valores numéricos acompanhados das unidades, se necessário. Todos os parâmetros são acompanhados de uma descrição para facilitar a sua compreensão.

Na figura 3.22 pode observar-se o planeamento elaborado para a interface de visualização da informação recolhida.

E-mail

Parameter	Value	Description
Is running	OK	All email services running
Check queue	2	There are 2 emails in qmail queue
Check smtp routes	OK	All smtp routes available
Check internet connection	OK	ip: 8.8.8.8 reachable
E-mail size limit	20011452 Bytes	Outgoing e-mail size limit is 250011452 Bytes
Qmail-forward limit	15125123 Bytes	Qmail-forward limit is 15125123 Bytes
Qmail ldap users	OK	User 'test' configured in ldap database
Log	OK	No errors found

Figura 3.22: Excerto do relatório relativo aos parâmetros do correio eletrónico

3.1.5.10 IM

No servidor de mensagens instantâneas, é necessário efetuar duas verificações que garantem o seu funcionamento - o estado de execução (*Status*) e a conectividade com o servidor (*Connectivity*). O estado de execução indica se o serviço se encontra a correr. Já a conectividade com o servidor, verificada a partir de uma ligação *telnet* para a porta correspondente, garante que o mesmo recebe e responde aos pedidos dos clientes.

Nas figuras 3.23 e 3.24 pode observar-se o planeamento elaborado para as interfaces de configuração e de visualização da informação recolhida, respetivamente.

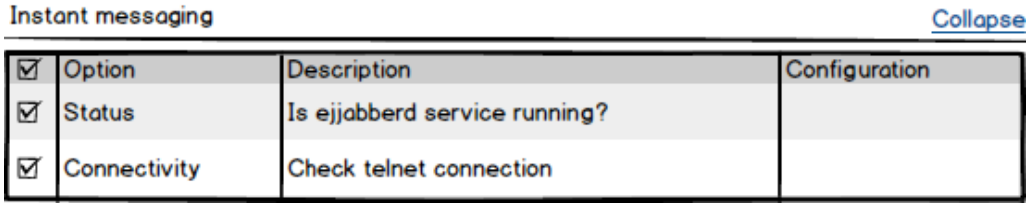


Figura 3.23: Área de selecção de parâmetros relativos às verificações do IM

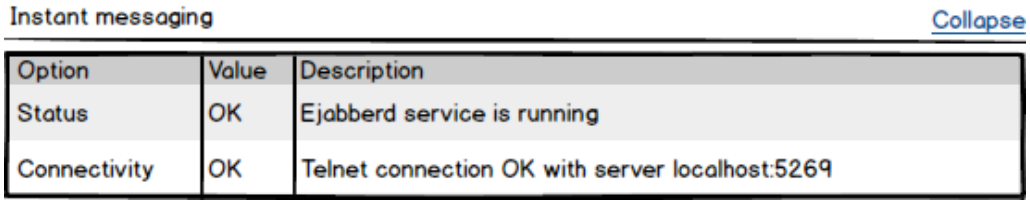


Figura 3.24: Excerto do relatório relativo aos parâmetros do IM

3.1.5.11 Servidor *web*

Já salientada a importância do servidor *web* para a solução IPBrick, importa saber quais os parâmetros que devem ser analisados. Na fase de planeamento foram considerados os seguintes parâmetros que caracterizam o funcionamento deste serviço:

- Estado do serviço (*Status*) - à semelhança de outros serviços, permite verificar se o servidor *web* se encontra em execução;
- Estatísticas (*Statistics*) - permite obter uma lista de valores relacionados com a utilização do servidor *web* de entre os quais se destacam o número de acessos, a utilização do Central processing unit (CPU), parâmetros da *cache*, entre outros. Estas indicações fornecem uma perspetiva geral da utilização do servidor *web* ;
- VirtualHosts - permite obter uma lista dos VirtualHosts configurados no servidor *web*;
- Ficheiros de configuração (*Configuration files*) - permite verificar a ocorrência de erros nos ficheiros de configuração do Apache. Se existirem erros, estes serão exibidos na descrição deste parâmetro.

Nas figuras 3.25 e 3.26 pode observar-se o planeamento elaborado para as interfaces de configuração e de visualização da informação recolhida, respetivamente.

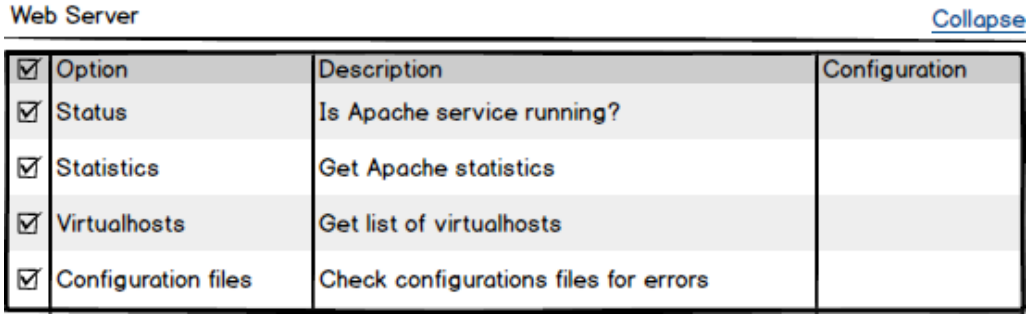


Figura 3.25: Área de selecção de parâmetros relativos às verificações do servidor *web*

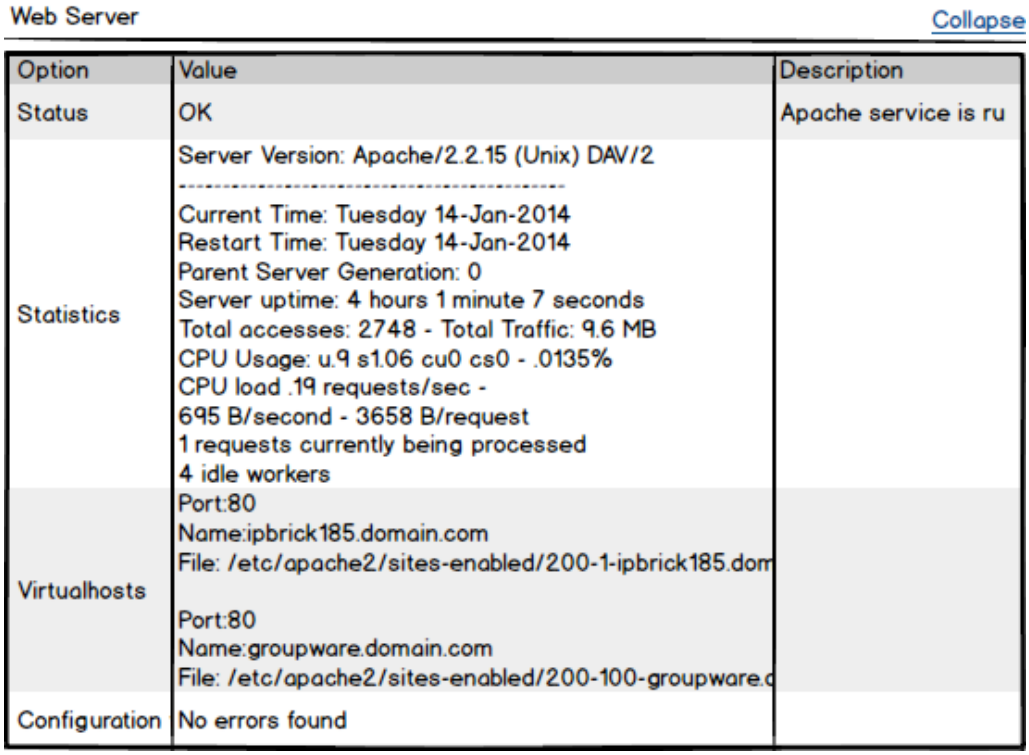


Figura 3.26: Excerto do relatório relativo aos parâmetros do servidor *web*

3.1.5.12 FAX

No servidor FAX são considerados os mesmos parâmetros que no servidor de IM. Tal como neste, a garantia de execução do serviço (*Status*) e conectividade *telnet* (*Connectivity*) permite certificar o seu correto funcionamento.

Nas figuras 3.27 e 3.28 pode observar-se o planeamento elaborado para as interfaces de configuração e de visualização da informação recolhida, respetivamente.

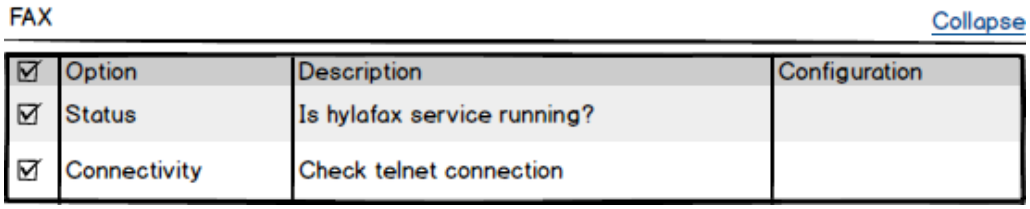


Figura 3.27: Área de selecção de parâmetros relativos às verificações do serviço de FAX

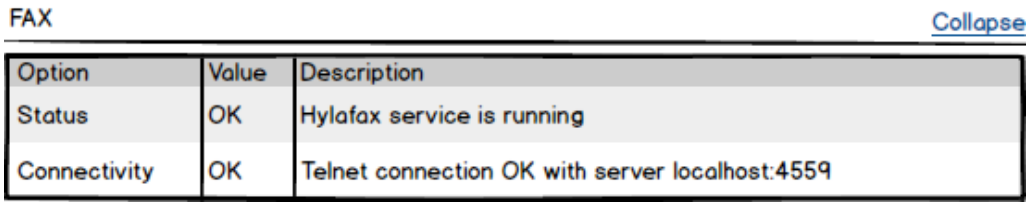


Figura 3.28: Excerto do relatório relativo aos parâmetros do serviço de FAX

3.1.5.13 Proxy

A análise ao servidor *proxy* engloba a avaliação de dois serviços - o Squid e o Dansguardian. Os parâmetros previstos nessa análise são os seguintes;

- Estado de execução do Squid (*Squid status*) - permite verificar o estado de execução do serviço Squid;
- Ocupação da memória *cache* no Squid (*Squid cache*) - permite obter o total de memória *cache* ocupada pelo Squid, o máximo definido e a percentagem de memória ocupada relativamente ao máximo;
- Ficheiros de configuração do Squid (*Configuration files*) - permite detetar a existência de erros nos ficheiros de configuração do Squid;
- Verificação se uma página não é bloqueada (*Check page*) - permite verificar se um Uniform Resource Locator (URL) definido é ou não bloqueado quando atravessa o *proxy*. Por vezes, as páginas são bloqueadas devido a filtros definidos no servidor *proxy*;
- Estado de execução do Dansguardian (*Dansguardian status*) - permite obter o estado de execução do Dansguardian;
- Últimos URL bloqueados pelo Dansguardian (*Dansguardian blocked urls*) - permite obter uma lista com os últimos URL bloqueados pelo Dansguardian e a razão do bloqueio.

Nas figuras 3.29 e 3.30 pode observar-se o planeamento elaborado para as interfaces de configuração e de visualização da informação recolhida, respetivamente.

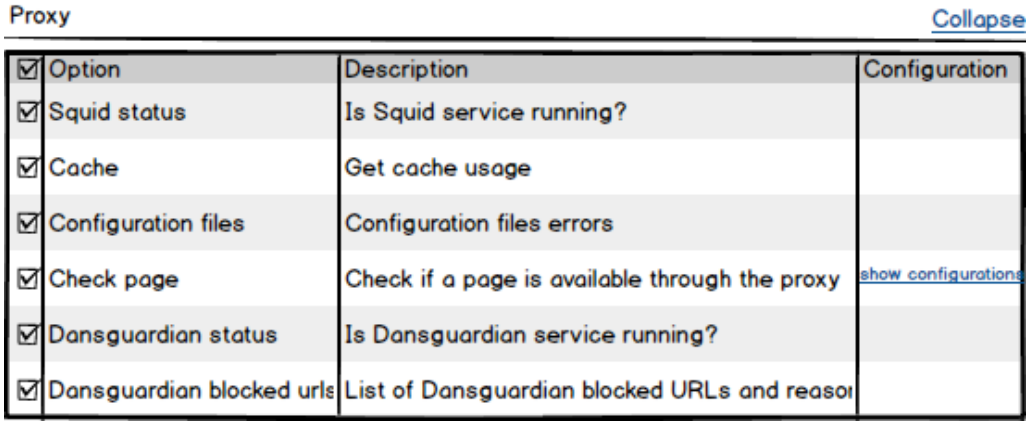


Figura 3.29: Área de selecção de parâmetros relativos às verificações do serviço de *proxy*

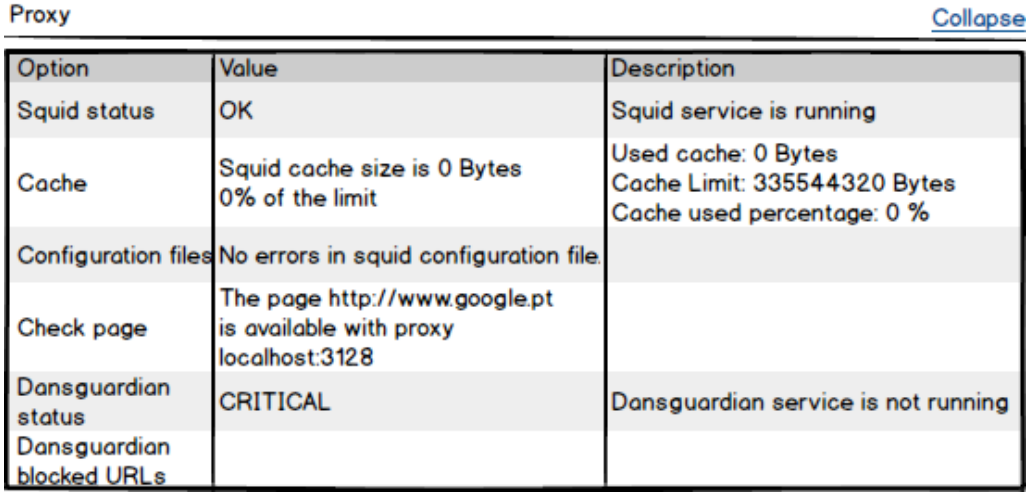


Figura 3.30: Excerto do relatório relativo aos parâmetros do serviço de *proxy*

3.1.6 Implementação

Definido o planeamento de todo o módulo, iniciou-se o desenvolvimento. Na primeira fase, deu-se prioridade aos *scripts*. Só depois de concluída esta fase se deu início à segunda fase que previa o desenvolvimento da interface *web*.

Durante a primeira fase, foi seguida a ordem de criticidade definida na tabela 3.1.1, sendo dada maior prioridade aos serviços com nível superior de criticidade.

Depois de desenvolvidos todos os *scripts*, iniciou-se o desenvolvimento da interface *web* seguindo o esboço mostrado nas secções anteriores.

O produto final apresenta as duas janelas já mencionadas na secção anterior - a interface de configuração e a interface de visualização da informação recolhida. A interface de configuração permite definir quais os parâmetros que serão analisados e a interface de visualização mostra as informações recolhidas e permite gerar um relatório no formato *pdf* que pode ser guardado ou enviado para o suporte IPBrick.

Na figura B.1, em anexo, pode observar-se a interface de configuração. De inicio, apenas os títulos dos serviços se encontram visíveis, mas, quando o utilizador selecciona a opção *Expand* ou simplesmente selecciona o serviço através da *checkbox* correspondente, as opções avançadas de configuração são mostradas. Nesta situação, o ecrã do utilizador será semelhante às figuras B.3 e B.4. Nestas figuras, que representam o mesmo ecrã, pode ainda observar-se que os serviços que não se encontram ativos no sistema operativo IPBrick não podem ser analisados, surgindo a informação de inatividade - neste caso, FAX e VPN. Os serviços podem encontrar-se inativos devido ao modelo de negócio oferecido por parte da IPBrick SA. A solução IPBrick é vendida como um conjunto de módulos. O cliente tem acesso aos módulos de acordo com a licença adquirida. E.g. O cliente pode não adquirir o módulo VPN, porque não pretende usufruir desse módulo, ficando com uma solução com um preço mais acessível.

Após a escolha, por parte do utilizador, das opções pretendidas, o botão *Run* - presente na figura B.4 - deve ser seleccionado. Após este gesto, o módulo executará, um a um, os *scripts* necessários para obter a informação pretendida. No decorrer deste processo, surge o ecrã das figuras B.5 e B.6 que fornece uma indicação ao utilizador da percentagem de tarefas executadas. Quando o processo finalizar, surge o relatório com todas as informações recolhidas, semelhante ao do anexo C, que será analisado com mais pormenor no capítulo 4.

Na figura C.1 podem observar-se as três opções disponíveis, no canto superior direito, que permitem voltar para o ecrã de configuração do módulo, enviar, por correio eletrónico, o relatório para o suporte IPBrick e descarregar o relatório no formato *pdf*. O relatório em formato *pdf* é semelhante ao que se encontra disponível no anexo D. As informações disponíveis neste relatório são as mesmas que estão disponíveis na interface *web* mas organizadas de uma forma um pouco diferente para que fosse possível gerar um relatório *pdf* legível e simples.

Apesar do planeamento realizado, foram implementadas algumas alterações que se deveram a novas funcionalidades requeridas pelos responsáveis da empresa. Assim, quando comparada a versão final com o planeamento, foram adicionadas as seguintes funcionalidades:

- Capacidade de deteção de processos que consomem mais recursos - é facultada ao utilizador uma lista dos processos que estão a utilizar maior percentagem de processamento, de RAM;
- Capacidade de deteção dos ficheiros de *log* mais antigos - é facultada ao utilizador uma lista dos ficheiros de *log* cuja data de criação seja anterior a uma data especificada;
- Capacidade de análise das interfaces de rede - permite ao utilizador obter o estado das interfaces de rede, bem como estatísticas de utilização. As estatísticas incluem dados importantes sobre os pacotes que fluem sobre as interfaces de rede que podem facilitar a deteção de ataques ao sistema;
- Capacidade de análise às estatísticas de rede - é facultada ao utilizador um conjunto de informações de estatísticas acerca da rede providenciadas pela ferramenta netstat [55].

Capítulo 4

Demonstração de resultados

Neste capítulo comprova-se o funcionamento da solução desenvolvida recorrendo a situações de erro propositadamente simuladas.

Devido às restrições da empresa e seus responsáveis, o teste do módulo desenvolvido numa IPBrick em produção não foi possível. Dadas estas limitações, optou-se pela simulação de erros num ambiente composto por uma IPBrick de teste.

Para facilitar a instalação do módulo, criou-se um pacote *debian*. Como a interface IPBrick prevê a instalação de pacotes *deb*, a instalação do módulo executou-se facilmente e sem problemas A.1.

Depois da instalação, as opções de configuração do módulo passaram a estar disponíveis no menu em *Advanced configurations* > *Monitoring* > *Troubleshooting*.

Seguidamente, simularam-se situações de modo a perceber se a deteção ocorria de forma acertada. Os resultados são apresentados nas secções seguintes.

4.1 Base de dados

Na figura 4.1 encontra-se um exemplo do serviço postgreSQL numa situação de funcionamento normal. As nove ligações à base de dados são explicadas com a execução do Kamailio que efetua constantes consultas.

Database		
Parameter	**Value**	**Description**
Database status	OK	Postgresql is running.
Database response	OK	PostgreSQL responds as expected
Number of active connections	9	Number of active connections: 9 Max number of connections: 200 Percentage of connections used: 4.5%
Disk usage	158937784 Bytes	The space used by all tables is 158937784 Bytes The largest table is dbdoc with 26764088 Bytes used.
Waiting Queries	0	Number of queries with WAITING status
Long time queries	0	There are 0 queries running for more than 50 ms

Figura 4.1: Situação sem erros no serviço de gestão de base de dados

A simulação de paragem do serviço não pode ser apresentada porque o utilizador não tem acesso à interface *web* nas situações em que a ligação à base de dados não é possível.

4.2 DNS

De forma a efetuar um teste à detecção de falhas no servidor DNS, forçou-se a paragem do serviço bind9 e, posteriormente, realizou-se a análise ao serviço. O resultado antes da paragem pode observar-se na figura 4.2. Na figura 4.3 encontra-se o resultado após a paragem do serviço.

DNS		
Parameter	**Value**	**Description**
DNS Status	OK	Dns service is running
Check resolution	OK	Dns resolution ok. For name 'cafe.domain.com' received ip '172.31.3.185' expected '172.31.3.185'

Figura 4.2: Situação sem erros no serviço de DNS

DNS		
Parameter	**Value**	**Description**
DNS Status	CRITICAL	The dns server is not running.
Check resolution	CRITICAL	

Figura 4.3: Situação com erros no serviço de DNS

Como era expectável, o facto do serviço não se encontrar em execução é assinalado no relatório com o valor *CRITICAL*.

4.3 DHCP

A deteção do correto funcionamento do serviço DHCP pode ser observada na figura 4.4. Posteriormente, forçou-se a paragem do serviço e obteve-se o alerta da figura 4.5.

DHCP		
Parameter	**Value**	**Description**
DHCP Status	OK	The dhcp server is running.

Figura 4.4: Situação sem erros no serviço de DHCP

DHCP		
Parameter	**Value**	**Description**
DHCP Status	CRITICAL	The dhcp server is not running.

Figura 4.5: Situação com erros no serviço de DHCP

4.4 LDAP

A afirmação do correto funcionamento do serviço openLDAP pode ser observada na figura 4.6. A situação de paragem forçada que foi simulada de seguida levou ao alerta da figura 4.7

LDAP		
Parameter	**Value**	**Description**
LDAP Status	OK	The ldap server and the ldap support services are running.

Figura 4.6: Situação sem erros no serviço LDAP

LDAP		
Parameter	**Value**	**Description**
LDAP Status	CRITICAL	The ldap server is not running.

Figura 4.7: Situação com erros no serviço LDAP

4.5 Firewall

Na figura 4.8 está representado um exemplo da informação obtida na análise às portas da *firewall*. Para o utilizador, surge uma lista de portas abertas com o protocolo e o serviço associados.

Firewall		
Parameter	**Value**	**Description**
Ports	> port 21 (protocol 'tcp') (service 'ftp') is open > port 22 (protocol 'tcp') (service 'ssh') is open > port 25 (protocol 'tcp') (service 'smtp') is open > port 53 (protocol 'tcp') (service 'domain') is open > port 80 (protocol 'tcp') (service 'http') is open > port 110 (protocol 'tcp') (service 'pop3') is open > port 111 (protocol 'tcp') (service 'rpcbind') is open > port 139 (protocol 'tcp') (service 'netbios-ssn') is open > port 143 (protocol 'tcp') (service 'imap') is open > port 389 (protocol 'tcp') (service 'ldap') is open > port 442 (protocol 'tcp') (service 'cvc_hostd') is open > port 443 (protocol 'tcp') (service 'https') is open > port 445 (protocol 'tcp') (service 'microsoft-ds') is open > port 628 (protocol 'tcp') (service 'qmqp') is open > port 953 (protocol 'tcp') (service 'rndc') is open > port 993 (protocol 'tcp') (service 'imaps') is open > port 995 (protocol 'tcp') (service 'pop3s') is open > port 1723 (protocol 'tcp') (service 'pptp') is open > port 2000 (protocol 'tcp') (service 'cisco-sccp') is open > port 2049 (protocol 'tcp') (service 'nfs') is open > port 3128 (protocol 'tcp') (service 'squid-http') is open > port 4369 (protocol 'tcp') (service 'epmd') is open > port 5038 (protocol 'tcp') (service 'unknown') is open > port 5222 (protocol 'tcp') (service 'xmpp-client') is open > port 5223 (protocol 'tcp') (service 'hpvirtgrp') is open > port 5269 (protocol 'tcp') (service 'xmpp-server') is open > port 5280 (protocol 'tcp') (service 'xmpp-bosh') is open > port 5281 (protocol 'tcp') (service 'unknown') is open > port 5432 (protocol 'tcp') (service 'postgresql') is open > port 5433 (protocol 'tcp') (service 'pyrrho') is open > port 8010 (protocol 'tcp') (service 'xmpp') is open > port 8731 (protocol 'tcp') (service 'unknown') is open > port 8888 (protocol 'tcp') (service 'sun-answerbook') is open > port 9571 (protocol 'tcp') (service 'unknown') is open > port 9572 (protocol 'tcp') (service 'unknown') is open > port 32854 (protocol 'tcp') (service 'unknown') is open > port 33150 (protocol 'tcp') (service 'unknown') is open > port 35468 (protocol 'tcp') (service 'unknown') is open > port 44208 (protocol 'tcp') (service 'unknown') is open > port 47271 (protocol 'tcp') (service 'unknown') is open > port 49320 (protocol 'tcp') (service 'unknown') is open	

Figura 4.8: Relatório obtido para a firewall

4.6 VoIP

Para comprovar a correta identificação dos problemas relacionados com o serviço de VoIP, executaram-se duas análises - uma com o serviço em execução e a responder corretamente, outra com o serviço parado. Na figura 4.9 está representado o resultado da primeira análise e na figura 4.10 está representado o resultada da segunda análise.

VOIP		
Parameter	**Value**	**Description**
VOIP Status	OK	The asterisk server is running.
Ports state	The SIP port is closed The SIPs port is closed	
Registered Phones	Number of registered phones: 0 Number of online phones: 0	
Objects	Number of objects: 4	Object Name: T38modem2 Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: toSobreiraOld Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: kamailio Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: toSobreira Object Type: peer Object flag: 0 Object RefCount: 2
Number of active calls	0	
Trunk	Connectable Trunks: 2 Unconnectable Trunks: 0 Unknown Trunks: 0	Connectable Trunks: @172.31.3.189 (SIP/2.0 200 OK) @172.31.3.188 (SIP/2.0 200 OK) Unconnectable Trunks: None Unknown Trunks: None
G729 License	Licensed channels: 0	Encoders in use: 0 Decoders in use: 0
Codecs	Codecs used now: No codecs in use.	
Agents	No Agents are configured in agents.conf	
Agents Online	No Agents are configured in agents.conf	
Do not disturb	0 results found.	
Telephony cards		

Figura 4.9: Situação sem erros no serviço de VoIP

VOIP		
Parameter	**Value**	**Description**
VOIP Status	CRITICAL	The asterisk server is not running.
Ports state	The SIP port is closed The SIPs port is closed	
Registered Phones	Number of registered phones: 0 Number of online phones: 0	
Objects	Number of objects: 0	
Number of active calls	0	
Trunk	Connectable Trunks: 2 Unconnectable Trunks: 0 Unknown Trunks: 0	Connectable Trunks: @172.31.3.189 (SIP/2.0 200 OK) @172.31.3.188 (SIP/2.0 200 OK) Unconnectable Trunks: None Unknown Trunks: None
G729 License	Licensed channels: 0	Encoders in use: 0 Decoders in use: 0
Codecs	Codecs used now: No codecs in use.	
Agents	Unable to connect to remote asterisk (does /var/run/asterisk/asterisk.ctl exist?)	
Agents Online	Unable to connect to remote asterisk (does /var/run/asterisk/asterisk.ctl exist?)	
Do not disturb	Unable to connect to remote asterisk (does /var/run/asterisk/asterisk.ctl exist?)	
Telephony cards		

Figura 4.10: Situação com erros no serviço de VoIP

Os dois relatórios das figuras 4.9 e 4.10 comprovam o funcionamento correto do módulo desenvolvido para a deteção do estado de execução dos serviços Asteriks e Kamailio. No entanto, pode ir-se mais longe e comprovar a correta deteção do estado das interfaces telefónicas. Para isso, criou-se um cenário de teste com uma IPBrick com uma placa de telefonia. Esta placa dispunha de duas portas PSTN. Ligaram-se as duas portas, uma à outra, e executou-se uma análise ao sistema

através do módulo, de seguida, cortou-se a ligação entre as duas portas e executou-se nova análise. Os resultados estão representados nas figuras 4.11 e 4.12, respetivamente.

VOIP		
Parameter	**Value**	**Description**
Telephony cards	PRI span 1/0: Up, Active PRI span 2/0: Up, Active	Port 1: Primary D-channel: 16 Status: Up, Active Switchtype: EuroISDN Type: CPE Remote type: Network Overlap Dial: 1 Logical Channel Mapping: 0 Timer and counter settings: N200: 3 N202: 3 K: 7 T200: 1000 T201: 1000 T202: 10000 T203: 10000 T303: 4000 T305: 30000 T308: 4000 T309: 6000 T312: 6000 T313: 4000 T316: -1 N316: 2 T-HOLD: 4000 T-RETRIEVE: 4000 T-RESPONSE: 4000 T-STATUS: 4000 T-ACTIVATE: 10000 T-DEACTIVATE: 4000 T-INTERROGATE: 4000 T-RETENTION: 30000 T-CCBS1: 4000 T-CCBS2: 2700000 T-CCBS3: 20000 T-CCBS4: 5000 T-CCBS5: 3600000 T-CCBS6: 3600000 T-CCNR2: 10800000 T-CCNR5: 11700000 T-CCNR6: 11700000 Q931 RX: 0 Q931 TX: 0 Q921 RX: 11371 Q921 TX: 11397 Q921 Outstanding: 0 (TEI=0) Total active-calls:0 global:0 CC records: Overlap Recv: Yes ---------- ---------- Port 2: Primary D-channel: 47 Status: Up, Active Switchtype: EuroISDN Type: Network Remote type: CPE Overlap Dial: 1 Logical Channel Mapping: 0 Timer and counter settings: N200: 3 N202: 3 K: 7 T200: 1000 T201: 1000 T202: 10000 T203: 10000 T303: 4000 T305: 30000 T308: 4000 T309: 6000 T312: 6000 T313: 4000 T316: -1 N316: 2 T-HOLD: 4000 T-RETRIEVE: 4000 T-RESPONSE: 4000 T-STATUS: 4000 T-ACTIVATE: 10000 T-DEACTIVATE: 4000 T-INTERROGATE: 4000 T-RETENTION: 30000 T-CCBS1: 4000 T-CCBS2: 2700000 T-CCBS3: 20000 T-CCBS4: 5000 T-CCBS5: 3600000 T-CCBS6: 3600000 T-CCNR2: 10800000 T-CCNR5: 11700000 T-CCNR6: 11700000 Q931 RX: 0 Q931 TX: 0 Q921 RX: 11376 Q921 TX: 11396 Q921 Outstanding: 0 (TEI=0) Total active-calls:0 global:0 CC records: Overlap Recv: Yes ----------

Figura 4.11: Placa de telefonia com as portas ligadas entre si

VOIP		
Parameter	**Value**	**Description**
Telephony cards	PRI span 1/0: In Alarm, Down, Active PRI span 2/0: In Alarm, Down, Active	---------- Port 1: Primary D-channel: 16 Status: In Alarm, Down, Active Switchtype: EuroISDN Type: CPE Remote type: Network Overlap Dial: 1 Logical Channel Mapping: 0 Timer and counter settings: N200: 3 N202: 3 K: 7 T200: 1000 T201: 1000 T202: 10000 T203: 10000 T303: 4000 T305: 30000 T308: 4000 T309: 6000 T312: 6000 T313: 4000 T316: -1 N316: 2 T-HOLD: 4000 T-RETRIEVE: 4000 T-RESPONSE: 4000 T-STATUS: 4000 T-ACTIVATE: 10000 T-DEACTIVATE: 4000 T-INTERROGATE: 4000 T-RETENTION: 30000 T-CCBS1: 4000 T-CCBS2: 2700000 T-CCBS3: 20000 T-CCBS4: 5000 T-CCBS5: 3600000 T-CCBS6: 3600000 T-CCNR2: 10800000 T-CCNR5: 11700000 T-CCNR6: 11700000 Q931 RX: 0 Q931 TX: 0 Q921 RX: 12983 Q921 TX: 20015 Q921 Outstanding: 0 (TEI=0) Total active-calls:0 global:0 CC records: Overlap Recv: Yes ---------- ---------- Port 2: Primary D-channel: 47 Status: In Alarm, Down, Active Switchtype: EuroISDN Type: Network Remote type: CPE Overlap Dial: 1 Logical Channel Mapping: 0 Timer and counter settings: N200: 3 N202: 3 K: 7 T200: 1000 T201: 1000 T202: 10000 T203: 10000 T303: 4000 T305: 30000 T308: 4000 T309: 6000 T312: 6000 T313: 4000 T316: -1 N316: 2 T-HOLD: 4000 T-RETRIEVE: 4000 T-RESPONSE: 4000 T-STATUS: 4000 T-ACTIVATE: 10000 T-DEACTIVATE: 4000 T-INTERROGATE: 4000 T-RETENTION: 30000 T-CCBS1: 4000 T-CCBS2: 2700000 T-CCBS3: 20000 T-CCBS4: 5000 T-CCBS5: 3600000

Figura 4.12: Placa de telefonia sem ligação nas portas

Como se pode observar nas figuras 4.11 e 4.12, o módulo indica a informação das portas de

telefonia e também o estado da ligação. Desta forma fica facilitada a identificação de problemas nas portas das placas de telefonia.

4.7 IM

À semelhança de outros serviços, no IM, simulou-se uma falha no serviço forçando a sua paragem. Antes da ação de parar sobre o serviço, o resultado da análise era o da figura 4.13. Depois desta ação, o resultado era o da figura 4.14.

Instant Message		
Parameter	**Value**	**Description**
IM Status	OK	Ejabberd service is running.
IM connectivity	OK	Ejabberd telnet connection ok.

Figura 4.13: Situação sem erros no serviço de IM

Depois de forçada a paragem, o resultado foi o da figura a seguir.

Instant Message		
Parameter	**Value**	**Description**
IM Status	CRITICAL	Ejabberd service is not running.
IM connectivity	CRITICAL	Ejabberd telnet connection error.

Figura 4.14: Situação com erros no serviço de IM

A figura captada no momento que precede a simulação sugere, como expectável, a execução do serviço. A figura captada no momento posterior à paragem forçada, confirma que o serviço está parado.

4.8 Email

Nas figuras seguintes está explicita a capacidade de deteção de falhas no serviço de correio eletrónico. Na figura 4.15 apresenta-se o resultado de uma análise com os servidores SMTP, POP e IMAP em pleno funcionamento. Na figura 4.16 surge o resultado de uma análise com a paragem forçada dos serviços Qmail, Courier-POP, Courier-IMAP, Courier-POP-SSL e Courier-IMAP-SSL. Com estas duas análises em ambiente controlado, pode afirmar-se que a deteção do estado de execução destes serviços ocorre como expectável.

E-mail		
Parameter	**Value**	**Description**
QMAIL status	All qmail processes are running.	
Queue	Number of queued messages: 1 Number of messages not pre processed: 1	
Smtp Routes	No SMTP routes configured	
Internet Connection	OK	Tested with ip 8.8.8.8
Send Limit	40000000 Bytes	
Forward Limit	20000000 Bytes	
LDAP users	User teste: User teste not found in Ldap	
POP status	OK	Pop service is running.
POPs status	OK	Pops service is running.
POP connectivity	OK	Pop connection ok.
IMAP status	OK	Imap service is running.
IMAPs status	OK	Imaps service is running.
IMAP connectivity	OK	Imap connection ok.

Figura 4.15: Situação sem erros no serviço de email

E-mail		
Parameter	**Value**	**Description**
QMAIL status	Qmail is not running correctly	
Queue	Number of queued messages: 1 Number of messages not pre processed: 1	
Smtp Routes	No SMTP routes configured	
Internet Connection	OK	Tested with ip 8.8.8.8
Send Limit	40000000 Bytes	
Forward Limit	20000000 Bytes	
LDAP users	User teste: User teste not found in Ldap	
POP status	CRITICAL	Pop service is not running.
POPs status	CRITICAL	Pops service is not running.
POP connectivity	CRITICAL	Pop connection error.
IMAP status	CRITICAL	Imap service is not running.
IMAPs status	CRITICAL	Imaps service is not running.
IMAP connectivity	CRITICAL	Imap connection error.

Figura 4.16: Situação com erros no serviço de email

Outra situação que foi simulada foi deteção do estado das rotas SMTP. Como referido anteriormente neste documento, as rotas SMTP permitem redireccionar as mensagens de email para outros servidores. As rotas configuradas no serviço Qmail são testadas, uma a uma, de forma a verificar a conetividade com o servidor de destino.

Na figura 4.17 encontra-se o resultado da análise ao sistema com uma rota em pleno funcionamento. Na figura 4.18 encontra-se o resultado ao sistema com uma rota para um destino cujo servidor SMTP não se encontra em execução.

E-mail		
Parameter	**Value**	**Description**
Smtp Routes	SMTP routes connectable: Route to sobreira.com SMTP routes unconnectable:	

Figura 4.17: Situação sem erros no serviço de email na deteção do estado das rotas SMTP

E-mail		
Parameter	**Value**	**Description**
Smtp Routes	SMTP routes connectable: None SMTP routes unconnectable: Route to: sobreira.com	

Figura 4.18: Situação com erros no serviço de email na deteção do estado das rotas SMTP

Uma vez mais, estas simulações confirmaram a capacidade de deteção de falhas por parte do módulo desenvolvido.

4.9 Servidor *web*

A capacidade de deteção do estado de execução do Apache é uma opção que, apesar de estar incluída no produto desenvolvido, não pode ser simulada e não deverá ser muito utilizada. De facto, quando o serviço Apache não se encontra em execução, não é possível aceder à interface *web* e, como consequência, não é possível aceder à interface do módulo desenvolvido.

Para a comprovação da correta deteção de falhas no servidor *web* optou-se pela simulação de erros nos ficheiros de configuração. Como se pode observar nas figuras 4.19 e 4.20, os erros foram corretamente divulgados.

Web Server		
Parameter	**Value**	**Description**
Apache status	OK	Apache service is running.
Apache statistics	Server Version: Apache/2.2.22 (Debian) proxy_html/3.0.1 mod_ssl/2.2.22 Server Built: Feb 1 2014 21:26:04 Current Time: Tuesday, 17-Jun-2014 17:18:27 BST Restart Time: Thursday, 05-Jun-2014 11:35:12 BST Parent Server Generation: 6 Server uptime: 12 days 5 hours 43 minutes 14 seconds Total accesses: 33536 - Total Traffic: 15.4 MB CPU Usage: u1556.34 s324.59 cu5.64 cs0 - .178% CPU load cache type: SHMCB, shared memory: 512000 bytes, current sessions: 0 subcaches: 32, indexes per subcache: 133 index usage: 0%, cache usage: 0% total sessions stored since starting: 0 total sessions expired since starting: 0 total (pre-expiry) sessions scrolled out of the cache: 0 total retrieves since starting: 0 hit, 76 miss total removes since starting: 0 hit, 0 miss	
Virtual hosts	Port: 80 Name: contacts.domain.com File: (/etc/apache2/sites-enabled/200-64-contacts.domain.com:6) Port: 80 Name: pgsqladmin.domain.com File: (/etc/apache2/sites-enabled/200-66-pgsqladmin.domain.com:6) Port: 80 Name: callmanager.domain.com File: (/etc/apache2/sites-enabled/200-71-callmanager.domain.com:6) Port: 80 Name: jwchat.domain.com File: (/etc/apache2/sites-enabled/200-72-jwchat.domain.com:6)	
Configuration files	Apache configuration files without errors.	

Figura 4.19: Situação sem erros no servidor *web*

Web Server		
Parameter	**Value**	**Description**
Apache status	OK	Apache service is running.
Apache statistics	Server Version: Apache/2.2.22 (Debian) proxy_html/3.0.1 mod_ssl/2.2.22 Server Built: Feb 1 2014 21:26:04 Current Time: Tuesday, 17-Jun-2014 17:57:18 BST Restart Time: Thursday, 05-Jun-2014 11:35:12 BST Parent Server Generation: 6 Server uptime: 12 days 6 hours 22 minutes 5 seconds Total accesses: 33655 - Total Traffic: 15.6 MB CPU Usage: u1578.97 s329.29 cu5.68 cs0 - .181% CPU load cache type: SHMCB, shared memory: 512000 bytes, current sessions: 0 subcaches: 32, indexes per subcache: 133 index usage: 0%, cache usage: 0% total sessions stored since starting: 0 total sessions expired since starting: 0 total (pre-expiry) sessions scrolled out of the cache: 0 total retrieves since starting: 0 hit, 82 miss total removes since starting: 0 hit, 0 miss	
Virtual hosts	No virtualhosts available.	
Configuration files	Apache configuration files with errors.	Syntax error on line 1 of /etc/apache2/sites-enabled/200-100-groupware.domain.com:

Figura 4.20: Situação com erros no servidor *web*

4.10 FAX

A deteção do correto funcionamento do serviço FAX pode ser observada nas figura 4.21 e 4.22. A primeira figura foi captada durante uma análise ao serviço em pleno funcionamento. A segunda figura foi captada durante uma análise que ocorreu posteriormente à paragem forçada do serviço. Como expectável, no segundo caso, o utilizador é informado do incorreto funcionamento do serviço.

Fax		
Parameter	**Value**	**Description**
Fax status	OK	Fax service is running.
Fax connectivity	OK	Fax telnet connection ok.
Fax queue	OK	Fax queue empty.

Figura 4.21: Situação sem erros no servidor FAX

Fax		
Parameter	**Value**	**Description**
Fax status	CRITICAL	Fax service is not running.
Fax connectivity	CRITICAL	Fax telnet connection error. Cannot connect to host 'localhost' and port '4559'
Fax queue	OK	Fax queue empty.

Figura 4.22: Situação com erros no servidor FAX

4.11 *Proxy*

Para confirmar a correta deteção de falhas no servidor *Proxy*, procedeu-se à paragem forçada do serviço Squid. A análise anterior à paragem e a análise a seguir à paragem podem ser comparadas nas figuras 4.23 e 4.24, respetivamente. Como se pode observar, o estado de execução do serviço foi lido corretamente em ambos os casos.

Proxy		
Parameter	**Value**	**Description**
Squid status	OK	Proxy service is running.
Squid cache	Squid cache size is 0 Bytes - 0% of the limit	Used cache: 0 Bytes Cache Limit: 335544320 Bytes Cache used percentage: 0 %
Squid check page	OK	The page http://www.google.pt is available with proxy localhost:3128
Dansguardian status	CRITICAL	Dansguardian service is not running.
Dansguardian blocked urls		
Configuration files	No errors in squid configuration file.	

Figura 4.23: Situação sem erros no servidor proxy

Proxy		
Parameter	**Value**	**Description**
Squid status	CRITICAL	Proxy service is not running.
Squid cache	Squid cache size is 0 Bytes - 0% of the limit	Used cache: 0 Bytes Cache Limit: 335544320 Bytes Cache used percentage: 0 %
Squid check page	CRITICAL	The page http://www.google.pt is not available with proxy localhost:3128
Dansguardian status	CRITICAL	Dansguardian service is not running.
Dansguardian blocked urls		
Configuration files	No errors in squid configuration file.	

Figura 4.24: Situação com erros no servidor proxy

Quanto ao Dansguardian, quando se começou o desenvolvimento do módulo previsto nesta dissertação, este era um serviço que vinha configurado por *default* na solução IPBrick. No entanto, houve alterações no sistema operativo que levaram à não inclusão do Dansguardian como serviço *default*. Assim, a opção de diagnóstico do Dansguardian continua a ser acessível no módulo, até porque é possível incluir este serviço opcionalmente, mas, quando o serviço não está presente surge a indicação de não execução.

4.12 *Considerações finais*

Todas as descrições anteriores se referem ao relatório gerado na interface *web*, no entanto, as mesmas informações também se encontram no ficheiro *pdf* que pode ser gerado (anexo D). Apesar do conteúdo ser o mesmo, a organização é ligeiramente diferente da interface *web* para facilitar o envia e, se necessário, a impressão do ficheiro.

De um modo geral, ficou comprovado o funcionamento correto das funcionalidades previstas para o módulo. Este módulo fornece a informação necessária para a identificação da fonte dos problemas. Fornece, ainda, indicadores que permitem definir o estado geral do sistema.

Capítulo 5

Conclusões

Neste capítulo é apresentada uma síntese do trabalho desenvolvido ao longo do semestre, referindo os resultados obtidos e as conclusões alcançadas. São também apresentadas as perspetivas de desenvolvimento futuro.

5.1 Síntese do trabalho desenvolvido

O trabalho desenvolvido levou à criação de uma ferramenta de deteção de falhas na solução IPBrick, que permite executar um *checkup* ao sistema e obter informação acerca do estado dos serviços e identificar os problemas.

A fase inicial do projeto focou-se no estudo do sistema operativo IPBrick, explorando todas as suas potencialidades. De seguida, foi efetuada uma pesquisa para obter as soluções do mercado que permitissem solucionar o problema principal, foram comparados os resultados e optou-se pelo desenvolvimento de um módulo IPBrick de raiz.

Após o definição da linguagem de programação, a metodologia e a estrutura a adoptar, iniciou-se a fase de desenvolvimento que culminou num produto que foi sujeito a testes que permitiram validar a sua utilidade.

5.2 Desenvolvimento futuro

O módulo desenvolvido serve como base de um projeto que deve ser desenvolvido a curto prazo e, por isso, está sujeito a possíveis atualizações. De facto, a escalabilidade e a robustez do módulo foi um fator que pesou na fase de planeamento e desenvolvimento.

Como melhorias futuras, deve destacar-se a continuação do desenvolvimento dos *scripts* para que exista a possibilidade de executar verificações em mais serviços. Estes novos *scripts* podem prever, não só uma ação passiva de recolha de dados mas também uma ação ativa de resolução de problemas.

Anexo A

Interface *web* final - Instalação

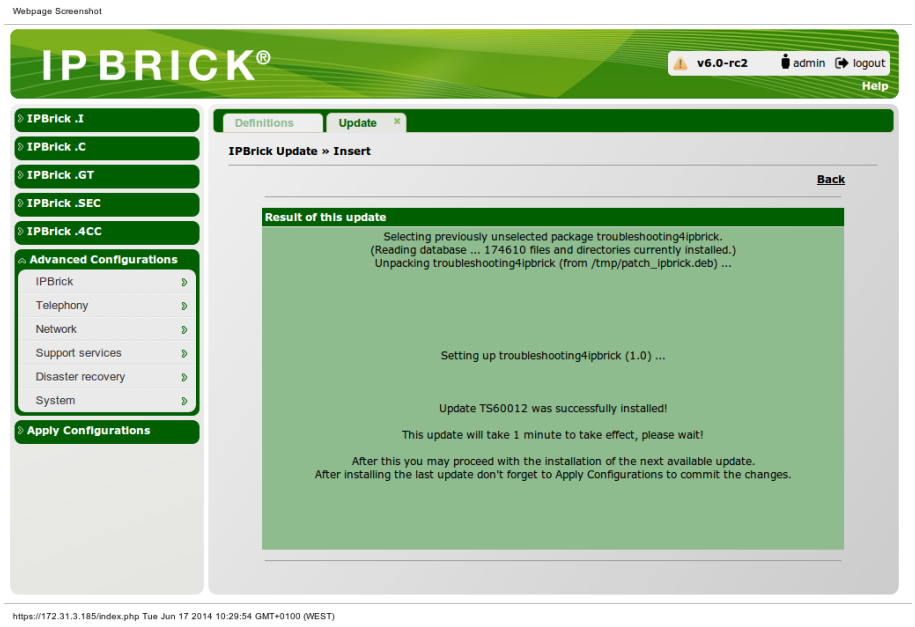


Figura A.1: Instalação do módulo *Troubleshooting4IPBrick*

Anexo B

Interface *web* final - Configuração

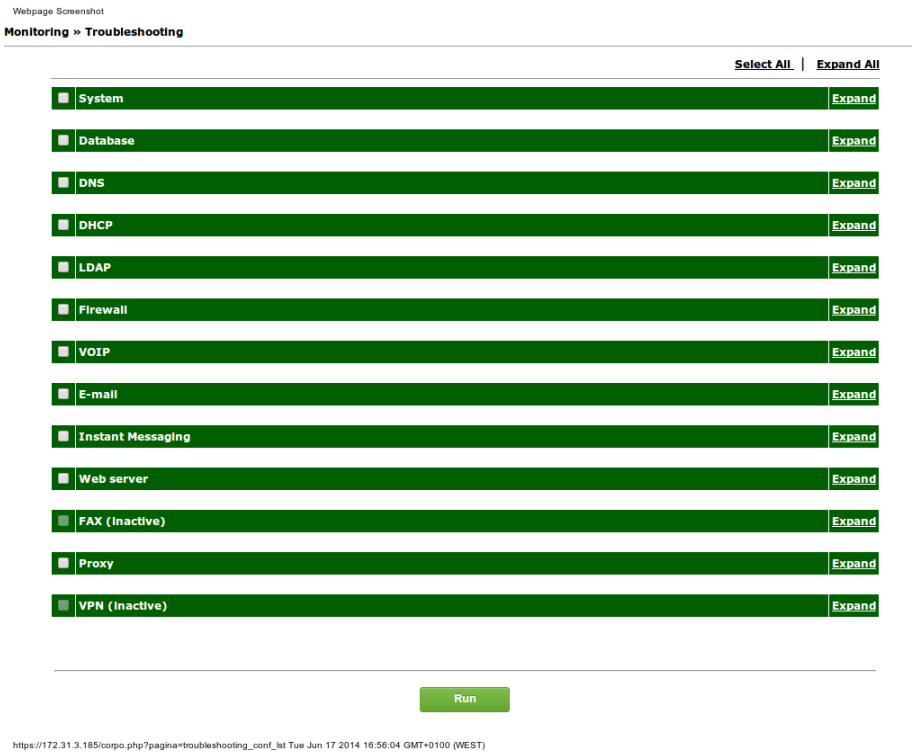


Figura B.1: Interface de configuração

Webpage Screenshot

Monitoring » Troubleshooting

Select All | Expand All

☑	**System**		Collapse
☑	Ram	Get Ram usage percentage	
☑	CPU	Get CPU usage percentage	
☑	Disk Usage	Get Disk usage percentage	
☑	Services analysis	Top cpu usage processes, top memory usage processes, top disk usage log files, get log files older than defined limit days	show configurations
☑	Network Interfaces	Network interfaces information	
☑	Network Statistics	Network statistics	
☑	**Database**		Collapse
☑	Status	Database status	
☑	Test query	Check if postgreSQL service is responding	
☑	Number of connections	Number of active connections to database	
☑	Disk usage	Disk used by postgreSQL tables	
☑	Waiting Queries	Number of queries with WAITING status	
☑	Long time queries	Number of queries running for more than limit time	show configurations
☑	**DNS**		Collapse
☑	DNS Status	DNS service status	
☑	Check name resolution	Check if name resolution is working	
☑	**DHCP**		Collapse
☑	DHCP Status	DHCP service status	
☑	**LDAP**		Collapse
☑	LDAP Status	LDAP service status	
☑	**Firewall**		Collapse
☑	Ports	Check if ports are open/closed	
☑	**VOIP**		Collapse
☑	VOIP Status	Asterisk service status	
☑	SIP ports	Check if sip ports are open	show configurations
☑	Registered Phones	Get registered phones	
☑	Objects	Show asterisk objects	
☑	Active calls	Number of active calls	
☑	Trunk	Check trunk connectivity	
☑	G729 License	Check G729 License	
☑	Codecs	Check used codecs	
☑	Agents	Show Asterisk agents	
☑	Agents online	Show Asterisk online agents	
☑	Do not disturb	Check if do not disturb option is enabled	
☑	Telephony cards	Telephony cards status	
☑	**E-mail**		Collapse
☑	QMAIL status	QMAIL service status	
☑	Queue	Number of e-mails in queue	
☑	Smtp Routes	Check smtp routes connectivity	
☑	Internet Connection	Check internet connection	show configurations
☑	Send Limit	Get size limit of sent mails	
☑	Forward Limit	Get size limit of forwarded mails	
☑	LDAP users	Check LDAP user information	show configurations
☑	Messages status		
☑	POP status		
☑	POPS status		
☑	POP connectivity		show configurations
☑	IMAP status		
☑	IMAPS status		
☑	IMAP connectivity		show configurations
☑	**Instant Messaging**		Collapse
☑	IM Status		
☑	IM connectivity		show configurations
☑	**Web server**		Collapse
☑	Apache status		
☑	Apache statistics		
☑	Virtual Hosts		
☑	Configuration files		
☐	**FAX (Inactive)**		Expand
☑	**Proxy**		Collapse
☑	SQUID status		
☑	Cache size		
☑	Check page		show configurations
☑	Dansguardian status		
☑	Dansguardian block reason		
☑	SQUID configuration files		
☐	**VPN (Inactive)**		Expand

Run

Figura B.3: Interface de configuração expandida (parte 1)

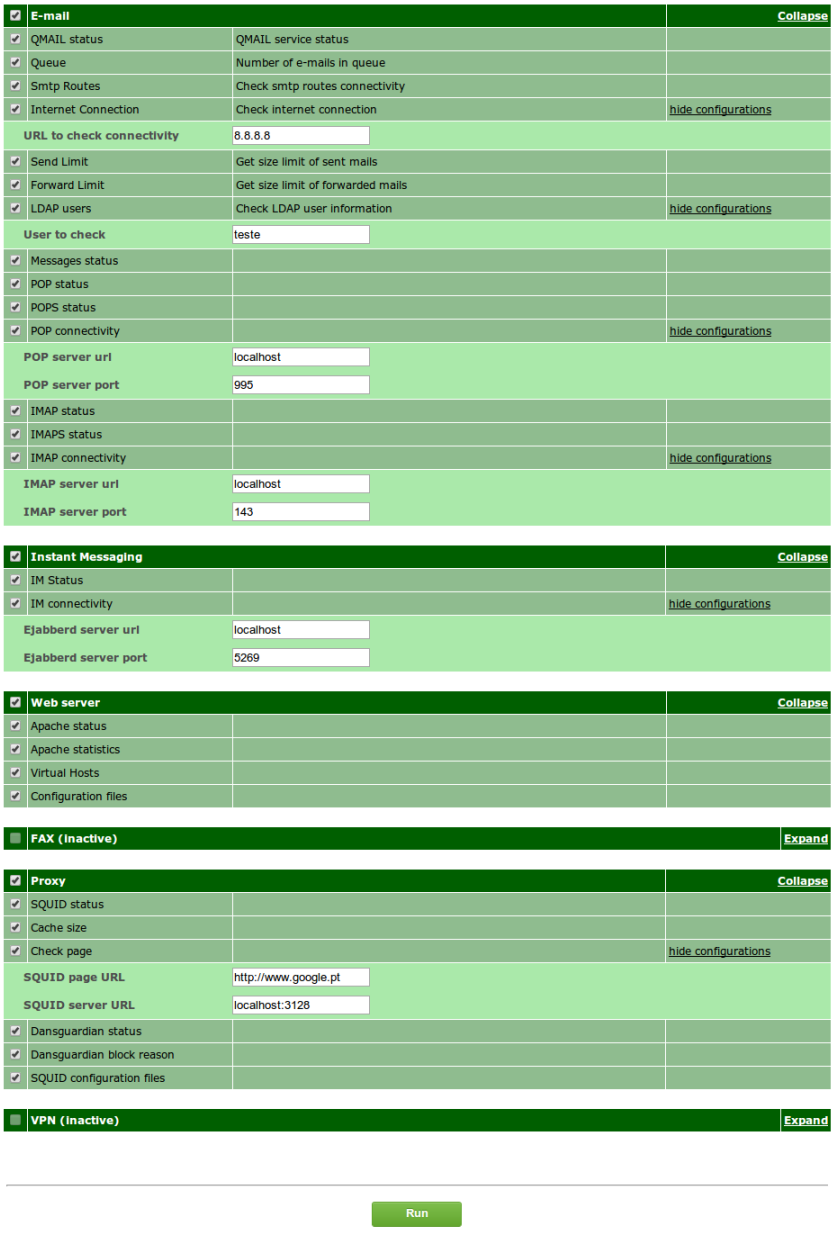


Figura B.4: Interface de configuração expandida (parte 2)

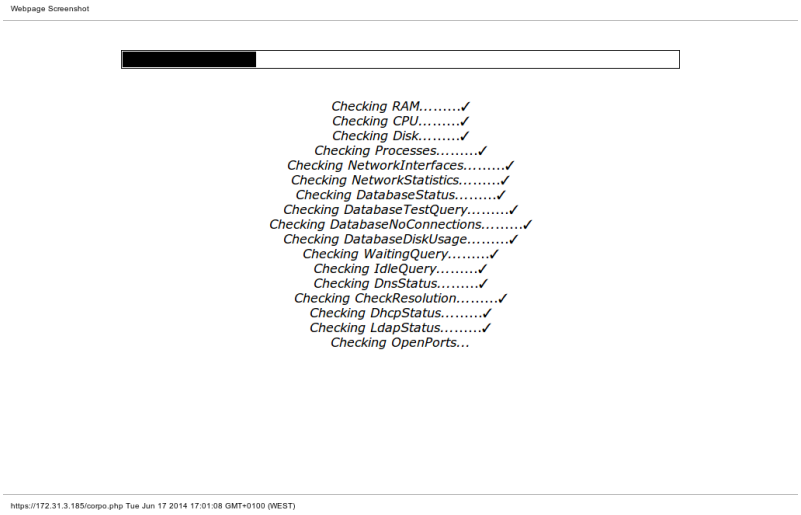


Figura B.5: Interface de indicação de espera enquanto são executados os *scripts* com cerca de 30% dos *scripts executados*

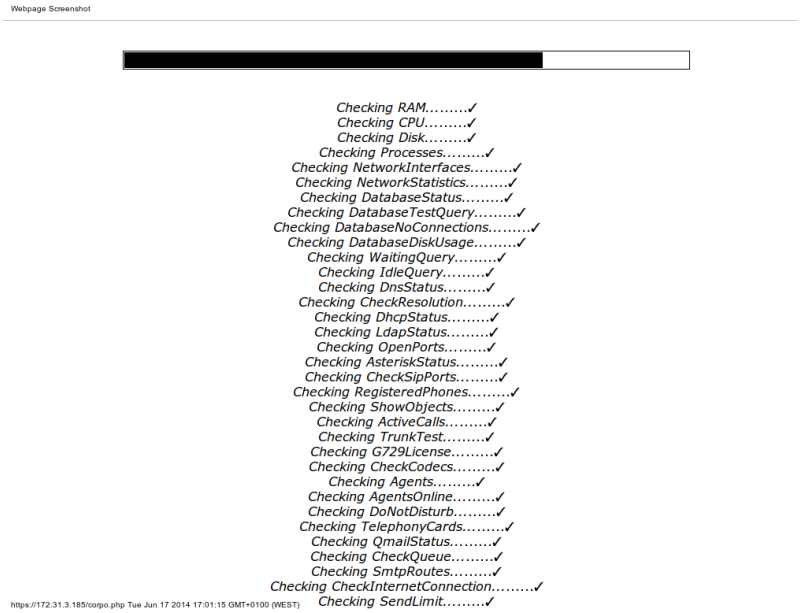


Figura B.6: Interface de indicação de espera enquanto são executados os *scripts* com cerca de 70% dos *scripts executados*

Anexo C

Interface *web* final - Visualização da informação obtida

Webpage Screenshot

Monitoring » Troubleshooting

Back | Send Report | Download Report

System		
Parameter	**Value**	**Description**
Ram	53.2 %	Percentage of used Ram
Cpu	User CPU: 1 % Idle CPU: 95 %	Percentage of processing power used and idle
Hard disk	Root partition: 21% Usr partition: 80% Var partition: 23% Opt partition: 45% Home1 partition: 2% Home2 partition: 2%	Percentage of disk space used by each partition
Top 10 CPU usage processes	---------- Pid: 9691 Cpu: 0.0 % Mem: 2.5 % Time: 2:40.47 Process: apache2 ---------- ---------- Pid: 96 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.00 Process: scsi_eh_1 ---------- ---------- Pid: 95 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.00 Process: scsi_eh_0 ---------- ---------- Pid: 9369 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.88 Process: kworker/0:0 ---------- ---------- Pid: 9 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.00 Process: khelper ---------- ---------- Pid: 84 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.00 Process: ata_sff ---------- ---------- Pid: 8 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.00 Process: cpuset ---------- ---------- Pid: 79 Cpu: 0.0 % Mem: 0.0 % Time: 0:00.47 Process: khubd ---------- ---------- Pid: 7302 Cpu: 0.0 % Mem: 0.2 % Time: 0:00.33 Process: psql ---------- ---------- Pid: 7293 Cpu: 0.0 % Mem: 0.1 % Time: 0:00.01 Process: bash ----------	
	---------- Pid: 25346 Cpu: 0.0 % Mem: 5.5 % Time: 0:40.26 Process: snort ---------- ---------- Pid: 5774 Cpu: 0.0 % Mem: 3.0 % Time: 4:28.08 Process: apache2 ---------- ---------- Pid: 15807 Cpu: 0.0 % Mem: 2.8 % Time: 3:40.55	

Figura C.1: Interface de visualização da informação recolhida - parte 1 de 7

Figura C.2: Interface de visualização da informação recolhida - parte 2 de 7

	Path: /var/log/qmail/smtpd/lock ---------- ---------- Size: 0b Date: Nov 14 2012 Path: /var/log/qmail/send/lock ----------
Network Interfaces	---------- Interface: eth0 Mac Address: de:ad:10:91:5a:8a IP Address: 172.31.3.185 Broadcast IP Address: 172.31.3.255 Packets received: 10187973 Packet errors received: 0 Dropped packets received: 0 Reception overruns: 0 Frame received: 0 Packets sent: 445028 Packet errors sent: 0 Dropped packets sent: 0 Transmission overruns: 0 Transmission carrier: 0 Transmission collisions: 0 Transmission queue: 1000 Bytes received: (1.0 GiB) Bytes sent: (91.0 MiB) ---------- ---------- Interface: eth1 Mac Address: de:ad:48:15:b7:a8 IP Address: 10.0.0.253 Broadcast IP Address: 10.0.0.255 Packets received: 756382 Packet errors received: 0 Dropped packets received: 0 Reception overruns: 0 Frame received: 0 Packets sent: 24822 Packet errors sent: 0 Dropped packets sent: 0 Transmission overruns: 0 Transmission carrier: 0 Transmission collisions: 0 Transmission queue: 1000 Bytes received: (174.1 MiB) Bytes sent: (2.0 MiB) ----------
Network Statistics	Ip: 23493098 total packets received 126 with invalid addresses 0 forwarded 0 incoming packets discarded 21763368 incoming packets delivered 19954384 requests sent out 1773 reassemblies required 745 packets reassembled ok Icmp: 160 ICMP messages received 0 input ICMP message failed. ICMP input histogram: destination unreachable: 72 echo requests: 27 echo replies: 61 162 ICMP messages sent 0 ICMP messages failed ICMP output histogram: destination unreachable: 59 echo request: 76 echo replies: 27 IcmpMsg: InType0: 61 InType3: 72 InType8: 27 OutType0: 27 OutType3: 59 OutType8: 76 Tcp: 438827 active connections openings 444918 passive connection openings 1953 failed connection attempts 281758 connection resets received 38 connections established 13365282 segments received 12263643 segments send out 12 segments retransmited 0 bad segments received. 1351427 resets sent Udp: 8122478 packets received 1 packets to unknown port received. 1030901 packet receive errors 6627421 packets sent RcvbufErrors: 101045 UdpLite: TcpExt: 89 invalid SYN cookies received 1184 resets received for embryonic SYN_RECV sockets 102 packets pruned from receive queue because of socket buffer overrun 161176 TCP sockets finished time wait in fast timer 688164 delayed acks sent 47 delayed acks further delayed because of locked socket Quick ack mode was activated 8 times 2470509 packets directly queued to recvmsg prequeue. 358678 bytes directly in process context from backlog 10951425 bytes directly received in process context from prequeue 3587269 packet headers predicted 573132 packets header predicted and directly queued to user 1123777 acknowledgments not containing data payload received 4658339 predicted acknowledgments 12 other TCP timeouts 1 times receiver scheduled too late for direct processing 7283 packets collapsed in receive queue due to low socket buffer 8 DSACKs sent for old packets 2 DSACKs received 276443 connections reset due to unexpected data 1823 connections reset due to early user close TCPBacklogDrop: 1 TCPDeferAcceptDrop: 6954 TCPChallengeACK: 5 IpExt: InMcastPkts: 20455 InBcastPkts: 1791466

Figura C.3: Interface de visualização da informação recolhida - parte 3 de 7

	OutBcastPkts: 27847 InOctets: 360691908 OutOctets: -395665423 InMcastOctets: 654560 InBcastOctets: 273007552 OutBcastOctets: 3211153

Database		
Parameter	**Value**	**Description**
Database status	OK	Postgresql is running.
Database response	OK	PostgreSQL responds as expected
Number of active connections	10	Number of active connections: 10 Max number of connections: 200 Percentage of connections used: 5%
Disk usage	159471804 Bytes	The space used by all tables is 159471804 Bytes The largest table is dbdoc with 26764088 Bytes used.
Waiting Queries	0	Number of queries with WAITING status
Long time queries	1	There are 1 queries running for more than 50 ms

DNS		
Parameter	**Value**	**Description**
DNS Status	OK	Dns service is running
Check resolution	OK	Dns resolution ok. For name 'cafe.domain.com' received ip '172.31.3.185' expected '172.31.3.185'

DHCP		
Parameter	**Value**	**Description**
DHCP Status	OK	The dhcp server is running.

LDAP		
Parameter	**Value**	**Description**
LDAP Status	OK	The ldap server and the ldap support services are running.

Firewall		
Parameter	**Value**	**Description**
Ports	> port 21 (protocol 'tcp') (service 'ftp') is open > port 22 (protocol 'tcp') (service 'ssh') is open > port 25 (protocol 'tcp') (service 'smtp') is open > port 53 (protocol 'tcp') (service 'domain') is open > port 80 (protocol 'tcp') (service 'http') is open > port 110 (protocol 'tcp') (service 'pop3') is open > port 111 (protocol 'tcp') (service 'rpcbind') is open > port 139 (protocol 'tcp') (service 'netbios-ssn') is open > port 143 (protocol 'tcp') (service 'imap') is open > port 389 (protocol 'tcp') (service 'ldap') is open > port 442 (protocol 'tcp') (service 'cvc_hostd') is open > port 443 (protocol 'tcp') (service 'https') is open > port 445 (protocol 'tcp') (service 'microsoft-ds') is open > port 628 (protocol 'tcp') (service 'qmqp') is open > port 953 (protocol 'tcp') (service 'rndc') is open > port 993 (protocol 'tcp') (service 'imaps') is open > port 995 (protocol 'tcp') (service 'pop3s') is open > port 1723 (protocol 'tcp') (service 'pptp') is open > port 2000 (protocol 'tcp') (service 'cisco-sccp') is open > port 2049 (protocol 'tcp') (service 'nfs') is open > port 3128 (protocol 'tcp') (service 'squid-http') is open > port 4369 (protocol 'tcp') (service 'epmd') is open > port 5038 (protocol 'tcp') (service 'unknown') is open > port 5222 (protocol 'tcp') (service 'xmpp-client') is open > port 5223 (protocol 'tcp') (service 'hpvirtgrp') is open > port 5269 (protocol 'tcp') (service 'xmpp-server') is open > port 5280 (protocol 'tcp') (service 'xmpp-bosh') is open > port 5281 (protocol 'tcp') (service 'unknown') is open > port 5432 (protocol 'tcp') (service 'postgresql') is open > port 5433 (protocol 'tcp') (service 'pyrrho') is open > port 8010 (protocol 'tcp') (service 'xmpp') is open > port 8731 (protocol 'tcp') (service 'unknown') is open > port 8888 (protocol 'tcp') (service 'sun-answerbook') is open > port 9571 (protocol 'tcp') (service 'unknown') is open > port 9572 (protocol 'tcp') (service 'unknown') is open > port 32854 (protocol 'tcp') (service 'unknown') is open > port 33150 (protocol 'tcp') (service 'unknown') is open > port 35468 (protocol 'tcp') (service 'unknown') is open > port 44208 (protocol 'tcp') (service 'unknown') is open > port 47271 (protocol 'tcp') (service 'unknown') is open > port 49320 (protocol 'tcp') (service 'unknown') is open	

VOIP		
Parameter	**Value**	**Description**
VOIP Status	OK	The asterisk server is running.
Ports state	The SIP port is closed The SIPs port is closed	
Registered Phones	Number of registered phones: 0 Number of online phones: 0	
Objects	Number of objects: 4	Object Name: T38modem2 Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: toSobreiraOld Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: kamailio Object Type: peer Object flag: 0 Object RefCount: 2 Object Name: toSobreira Object Type: peer Object flag: 0 Object RefCount: 2
Number of active calls	0	
Trunk	Connectable Trunks: 2 Unconnectable Trunks: 0	Connectable Trunks: @172.31.3.189 (SIP/2.0 200 OK) @172.31.3.188 (SIP/2.0 200 OK) Unconnectable Trunks:

Figura C.4: Interface de visualização da informação recolhida - parte 4 de 7

	Unknown Trunks: 0	None Unknown Trunks: None
G729 License	Licensed channels: 0	Encoders in use: 0 Decoders in use: 0
Codecs	Codecs used now: No codecs in use.	
Agents	No Agents are configured in agents.conf	
Agents Online	No Agents are configured in agents.conf	
Do not disturb	0 results found.	
Telephony cards		

E-mail		
Parameter	**Value**	**Description**
QMAIL status	All qmail processes are running.	
Queue	Number of queued messages: 1 Number of messages not pre processed: 1	
Smtp Routes	No SMTP routes configured	
Internet Connection	OK	Tested with ip 8.8.8.8
Send Limit	40000000 Bytes	
Forward Limit	20000000 Bytes	
LDAP users	User teste: User teste not found in Ldap	
	---------- Message number 4357 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1401464710_qp_14814_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4478 From: To: remote config.backup@ipbrick.com Status: success: June_5,_2014,_10:12_am_-_16529_- _DIR:/tmp/emails/mail1401959527.61463700/_Fim/did_0+0+1/ ---------- ---------- Message number 4493 From: To: remote luis.borges1991@gmail.com Status: success: 173.194.67.27_accepted_message./Remote_host_said:_250_2.0.0_OK_1401958991_pg7si10083798wjb.56_- _gsmtp/ ---------- ---------- Message number From: To: Status: success: 173.194.67.27_accepted_message./Remote_host_said:_250_2.0.0_OK_1401958991_eb1si40071239wic.29_-_gsmtp/ ---------- ---------- Message number 4492 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1401963745_qp_32752_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4491 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402053158_qp_28487_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4491 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402060798_qp_11958_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4491 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402062150_qp_15718_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402593515_qp_21123_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402594110_qp_22302_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402595569_qp_26226_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402595877_qp_26778_by_proxy2.iportalmais.pt/	

Figura C.5: Interface de visualização da informação recolhida - parte 5 de 7

Messages status	---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402596098_qp_27288_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402995097_qp_10994_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402995647_qp_11651_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402996413_qp_12907_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402996639_qp_13237_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402996688_qp_13353_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402996907_qp_13576_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402997055_qp_13694_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402997268_qp_13973_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402997535_qp_14473_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402997922_qp_15094_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402998799_qp_16677_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402999029_qp_16976_by_proxy2.iportalmais.pt/ ---------- ---------- Message number 4489 From: To: remote config.backup@ipbrick.com Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402999412_qp_17515_by_proxy2.iportalmais.pt/ ----------	
POP status	OK	Pop service is running.
POPs status	OK	Pops service is running.
POP connectivity	OK	Pop connection ok.
IMAP status	OK	Imap service is running.
IMAPs status	OK	Imaps service is running.
IMAP connectivity	OK	Imap connection ok.

Figura C.6: Interface de visualização da informação recolhida - parte 6 de 7

Instant Message		
Parameter	**Value**	**Description**
IM Status	OK	Ejabberd service is running.
IM connectivity	OK	Ejabberd telnet connection ok.

Web Server		
Parameter	**Value**	**Description**
Apache status	OK	Apache service is running.
Apache statistics	Server Version: Apache/2.2.22 (Debian) proxy_html/3.0.1 mod_ssl/2.2.22 Server Built: Feb 1 2014 21:26:04 Current Time: Tuesday, 17-Jun-2014 17:01:16 BST Restart Time: Thursday, 05-Jun-2014 11:35:12 BST Parent Server Generation: 6 Server uptime: 12 days 5 hours 26 minutes 4 seconds Total accesses: 33478 - Total Traffic: 15.0 MB CPU Usage: u1545.69 s322.04 cu5.58 cs0 - .177% CPU load cache type: SHMCB, shared memory: 512000 bytes, current sessions: 0 subcaches: 32, indexes per subcache: 133 index usage: 0%, cache usage: 0% total sessions stored since starting: 0 total sessions expired since starting: 0 total (pre-expiry) sessions scrolled out of the cache: 0 total retrieves since starting: 0 hit, 73 miss total removes since starting: 0 hit, 0 miss	
Virtual hosts	Port: 80 Name: lborges185.domain.com File: (/etc/apache2/sites-enabled/200-1-lborges185.domain.com:42) Port: 80 Name: groupware.domain.com File: (/etc/apache2/sites-enabled/200-100-groupware.domain.com:37) Port: 80 Name: wpad.domain.com File: (/etc/apache2/sites-enabled/200-112-wpad.domain.com:6) Port: 80 Name: ipbrick4cc.domain.com File: (/etc/apache2/sites-enabled/200-118-ipbrick4cc.domain.com:18) Port: 80 Name: broker.domain.com File: (/etc/apache2/sites-enabled/200-119-broker.domain.com:6) Port: 80 Name: cafe.domain.com File: (/etc/apache2/sites-enabled/200-128-cafe.domain.com:37) Port: 80 Name: ucoip.domain.com File: (/etc/apache2/sites-enabled/200-200-ucoip.domain.com:6) Port: 80 Name: contacts.domain.com File: (/etc/apache2/sites-enabled/200-64-contacts.domain.com:6) Port: 80 Name: pgsqladmin.domain.com File: (/etc/apache2/sites-enabled/200-66-pgsqladmin.domain.com:6) Port: 80 Name: callmanager.domain.com File: (/etc/apache2/sites-enabled/200-71-callmanager.domain.com:6) Port: 80 Name: jwchat.domain.com File: (/etc/apache2/sites-enabled/200-72-jwchat.domain.com:6) Port: 80 Name: webphone.domain.com File: (/etc/apache2/sites-enabled/200-73-webphone.domain.com:6) Port: 80 Name: callstatistics.domain.com File: (/etc/apache2/sites-enabled/200-74-callstatistics.domain.com:28) Port: 443 Name: lborges185.domain.com File: (/etc/apache2/sites-enabled/200-1-lborges185.domain.com:6) Port: 443 Name: groupware.domain.com File: (/etc/apache2/sites-enabled/200-100-groupware.domain.com:6) Port: 443 Name: ipbrick4cc.domain.com File: (/etc/apache2/sites-enabled/200-118-ipbrick4cc.domain.com:6) Port: 443 Name: cafe.domain.com File: (/etc/apache2/sites-enabled/200-128-cafe.domain.com:6) Port: 443 Name: callstatistics.domain.com File: (/etc/apache2/sites-enabled/200-74-callstatistics.domain.com:6)	
Configuration files	Apache configuration files without errors.	

Proxy		
Parameter	**Value**	**Description**
Squid status	OK	Proxy service is running.
Squid cache	Squid cache size is 0 Bytes - 0% of the limit	Used cache: 0 Bytes Cache Limit: 335544320 Bytes Cache used percentage: 0 %
Squid check page	OK	The page http://www.google.pt is available with proxy localhost:3128
Dansguardian status	CRITICAL	Dansguardian service is not running.
Dansguardian blocked urls		
Configuration files	No errors in squid configuration file.	

Figura C.7: Interface de visualização da informação recolhida - parte 7 de 7

Anexo D

Relatório exemplo

System

> Parameter: Ram
>> Value:
>>> 57.96 %
>> Description:
>>> Percentage of used Ram
> Parameter: Cpu
>> Value:
>>> User CPU: 3 %
>>> Idle CPU: 95 %
>> Description:
>>> Percentage of processing power used and idle
> Parameter: Hard disk
>> Value:
>>> Root partition: 13%
>>> Usr partition: 65%
>>> Var partition: 5%
>>> Opt partition: 31%
>>> Home1 partition: 1%
>>> Home2 partition: 1%
>>>
>> Description:
>>> Percentage of disk space used by each partition
> Parameter: Top 10 CPU usage processes
>> Value:
>>> ---
>>> Pid: 97
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.00
>>> Process: khubd
>>> ---
>>>
>>> ---
>>> Pid: 8
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.60
>>> Process: migration/1
>>> ---
>>>
>>> ---
>>> Pid: 732
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.00
>>> Process: firewire
>>> ---
>>>
>>> ---
>>> Pid: 7
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.20
>>> Process: watchdog/0
>>> ---
>>>
>>> ---
>>> Pid: 6
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.15
>>> Process: migration/0
>>> ---
>>>
>>> ---
>>> Pid: 5372
>>> Cpu: 0.0 %
>>> Mem: 0.4 %
>>> Time: 0:00.02
>>> Process: postgres
>>> ---
>>>
>>> ---
>>> Pid: 5345
>>> Cpu: 0.0 %
>>> Mem: 0.3 %
>>> Time: 0:00.00
>>> Process: postgres
>>> ---
>>>
>>> ---
>>> Pid: 5341
>>> Cpu: 0.0 %
>>> Mem: 0.3 %
>>> Time: 0:00.01

>>> Process: postgres
>>> ----------
>>>
>>> ----------
>>> Pid: 5323
>>> Cpu: 0.0 %
>>> Mem: 0.0 %
>>> Time: 0:00.49
>>> Process: astcanary
>>> ----------
>>>
>>> ----------
>>> Pid: 5322
>>> Cpu: 0.0 %
>>> Mem: 1.6 %
>>> Time: 1:54.22
>>> Process: asterisk
>>> ----------
>>>
>>>
>> Description:

> Parameter: Top 10 Memory usage processes
>> Value:
>>> ----------
>>> Pid: 25210
>>> Cpu: 0.0 %
>>> Mem: 2.9 %
>>> Time: 3:37.32
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 12606
>>> Cpu: 0.0 %
>>> Mem: 2.8 %
>>> Time: 3:25.26
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 12605
>>> Cpu: 0.0 %
>>> Mem: 2.8 %
>>> Time: 3:41.10
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 12603
>>> Cpu: 0.0 %
>>> Mem: 2.8 %
>>> Time: 3:48.91
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 12604
>>> Cpu: 0.0 %
>>> Mem: 2.7 %
>>> Time: 3:07.91
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 12602
>>> Cpu: 0.0 %
>>> Mem: 2.7 %
>>> Time: 3:24.31
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 3158
>>> Cpu: 0.0 %
>>> Mem: 2.5 %
>>> Time: 0:05.83
>>> Process: beam
>>> ----------
>>>
>>> ----------
>>> Pid: 25192
>>> Cpu: 0.0 %
>>> Mem: 2.5 %
>>> Time: 2:04.80
>>> Process: apache2
>>> ----------
>>>

>>> ----------
>>> Pid: 25157
>>> Cpu: 0.0 %
>>> Mem: 2.5 %
>>> Time: 2:46.36
>>> Process: apache2
>>> ----------
>>>
>>> ----------
>>> Pid: 25080
>>> Cpu: 0.0 %
>>> Mem: 2.5 %
>>> Time: 2:21.46
>>> Process: apache2
>>> ----------
>>>
>>>
>> Description:

> Parameter: Top 10 Disk usage files
>> Value:
>>> ----------
>>> Size: 960Kb
>>> Directory: /var/log/dpkg.log
>>> ----------
>>>
>>> ----------
>>> Size: 600Kb
>>> Directory: /var/log/lastlog
>>> ----------
>>>
>>> ----------
>>> Size: 540Kb
>>> Directory: /var/log/ejabberd
>>> ----------
>>>
>>> ----------
>>> Size: 424Kb
>>> Directory: /var/log/squidguard
>>> ----------
>>>
>>> ----------
>>> Size: 204Kb
>>> Directory: /var/log/apt
>>> ----------
>>>
>>> ----------
>>> Size: 96Kb
>>> Directory: /var/log/groupware
>>> ----------
>>>
>>> ----------
>>> Size: 68Kb
>>> Directory: /var/log/faillog
>>> ----------
>>>
>>> ----------
>>> Size: 52Kb
>>> Directory: /var/log/aptitude
>>> ----------
>>>
>>> ----------
>>> Size: 48Kb
>>> Directory: /var/log/dmesg.0
>>> ----------
>>>
>>> ----------
>>> Size: 48Kb
>>> Directory: /var/log/dmesg
>>> ----------
>>>
>>>
>> Description:

> Parameter: Oldest log files
>> Value:
>>> ----------
>>> Size: 40Kb
>>> Date: Feb 4 13:50
>>> Path: /var/log/bootstrap.log
>>> ----------
>>>
>>> ----------
>>> Size: 0b
>>> Date: Nov 14 2012
>>> Path: /var/log/qmail/send/lock
>>> ----------
>>>

>>> ----------
>>> Size: 0b
>>> Date: Nov 14 2012
>>> Path: /var/log/qmail/smtpd/lock
>>> ----------
>>>
>>> ----------
>>> Size: 0b
>>> Date: Nov 14 2012
>>> Path: /var/log/qmail/qmqpd/lock
>>> ----------
>>>
>>>
>> Description:

> Parameter: Network Interfaces
>> Value:
>>> ----------
>>> Interface: eth0
>>> Mac Address: 00:90:fb:33:1c:d0
>>> IP Address: 172.31.3.179
>>> Broadcast IP Address: 172.31.3.255
>>> Packets received: 551069
>>> Packet errors received: 0
>>> Dropped packets received: 0
>>> Reception overruns: 0
>>> Frame received: 0
>>> Packets sent: 64828
>>> Packet errors sent: 0
>>> Dropped packets sent: 0
>>> Transmission overruns: 0
>>> Transmission carrier: 0
>>> Transmission collisions: 0
>>> Transmission queue: 1000
>>> Bytes received: (94.3 MiB)
>>> Bytes sent: (22.2 MiB)
>>> ----------
>>> ----------
>>> Interface: eth1
>>> Mac Address: 00:90:fb:33:1c:d1
>>> IP Address: 10.0.0.253
>>> Broadcast IP Address: 10.0.0.255
>>> Packets received: 0
>>> Packet errors received: 0
>>> Dropped packets received: 0
>>> Reception overruns: 0
>>> Frame received: 0
>>> Packets sent: 0
>>> Packet errors sent: 0
>>> Dropped packets sent: 0
>>> Transmission overruns: 0
>>> Transmission carrier: 0
>>> Transmission collisions: 0
>>> Transmission queue: 1000
>>> Bytes received: (0.0 B)
>>> Bytes sent: (0.0 B)
>>> ----------
>>>
>> Description:

> Parameter: Network Statistics
>> Value:
>>> Ip:
>>> 5384822 total packets received
>>> 0 forwarded
>>> 0 incoming packets discarded
>>> 5366741 incoming packets delivered
>>> 5272185 requests sent out
>>> Icmp:
>>> 30 ICMP messages received
>>> 0 input ICMP message failed.
>>> ICMP input histogram:
>>> destination unreachable: 3
>>> echo requests: 2
>>> echo replies: 25
>>> 34 ICMP messages sent
>>> 0 ICMP messages failed
>>> ICMP output histogram:
>>> destination unreachable: 4
>>> echo request: 28
>>> echo replies: 2
>>> IcmpMsg:
>>> InType0: 25
>>> InType3: 3
>>> InType8: 2
>>> OutType0: 2
>>> OutType3: 4
>>> OutType8: 28

>>> Tcp:
>>> 118829 active connections openings
>>> 119876 passive connection openings
>>> 844 failed connection attempts
>>> 98010 connection resets received
>>> 82 connections established
>>> 4710224 segments received
>>> 3907734 segments send out
>>> 187 segments retransmited
>>> 0 bad segments received.
>>> 889757 resets sent
>>> Udp:
>>> 745313 packets received
>>> 1 packets to unknown port received.
>>> 0 packet receive errors
>>> 568986 packets sent
>>> UdpLite:
>>> TcpExt:
>>> 160 invalid SYN cookies received
>>> 780 resets received for embryonic SYN_RECV sockets
>>> 28677 TCP sockets finished time wait in fast timer
>>> 203909 delayed acks sent
>>> 1 delayed acks further delayed because of locked socket
>>> 850018 packets directly queued to recvmsg prequeue.
>>> 15 bytes directly in process context from backlog
>>> 1445728 bytes directly received in process context from prequeue
>>> 1037342 packet headers predicted
>>> 126513 packets header predicted and directly queued to user
>>> 293402 acknowledgments not containing data payload received
>>> 1285774 predicted acknowledgments
>>> 41 other TCP timeouts
>>> 1 DSACKs received
>>> 82889 connections reset due to unexpected data
>>> 7658 connections reset due to early user close
>>> 28 connections aborted due to timeout
>>> TCPDeferAcceptDrop: 1131
>>> TCPChallengeACK: 722
>>> IpExt:
>>> InBcastPkts: 89184
>>> OutBcastPkts: 1513
>>> InOctets: 946449403
>>> OutOctets: 896842953
>>> InBcastOctets: 9643014
>>> OutBcastOctets: 174684
>>>
>> Description:

Database

> Parameter: Database status
>> Value:
>>> OK
>> Description:
>>> Postgresql is running.
> Parameter: Database response
>> Value:
>>> OK
>> Description:
>>> PostgreSQL responds as expected
> Parameter: Number of active connections
>> Value:
>>> 32
>> Description:
>>> Number of active connections: 32
>>> Max number of connections: 200
>>> Percentage of connections used: 16%
>>>
> Parameter: Disk usage
>> Value:
>>> 158644412 Bytes
>> Description:
>>> The space used by all tables is 158644412 Bytes
>>> The largest table is dbdoc with 26764088 Bytes used.
> Parameter: Waiting Queries
>> Value:
>>> 0
>> Description:
>>> Number of queries with WAITING status
> Parameter: Long time queries
>> Value:
>>> 23
>> Description:
>>> There are 23 queries running for more than 50 ms

DNS

> Parameter: DNS Status

>> Value:
>>> OK
>> Description:
>>> Dns service is running
> Parameter: Check resolution
>> Value:
>>> OK
>> Description:
>>> Dns resolution ok. For name 'cafe.domain.com' received ip '172.31.3.179' expected '172.31.3.179'

DHCP

> Parameter: DHCP Status
>> Value:
>>> OK
>> Description:
>>> The dhcp server is running.

LDAP

> Parameter: LDAP Status
>> Value:
>>> OK
>> Description:
>>> The ldap server and the ldap support services are running.

Firewall

> Parameter: Ports
>> Value:
>>> > port 21 (protocol 'tcp') (service 'ftp') is open
>>> > port 22 (protocol 'tcp') (service 'ssh') is open
>>> > port 25 (protocol 'tcp') (service 'smtp') is open
>>> > port 53 (protocol 'tcp') (service 'domain') is open
>>> > port 80 (protocol 'tcp') (service 'http') is open
>>> > port 110 (protocol 'tcp') (service 'pop3') is open
>>> > port 111 (protocol 'tcp') (service 'rpcbind') is open
>>> > port 139 (protocol 'tcp') (service 'netbios-ssn') is open
>>> > port 143 (protocol 'tcp') (service 'imap') is open
>>> > port 389 (protocol 'tcp') (service 'ldap') is open
>>> > port 442 (protocol 'tcp') (service 'cvc_hostd') is open
>>> > port 443 (protocol 'tcp') (service 'https') is open
>>> > port 445 (protocol 'tcp') (service 'microsoft-ds') is open
>>> > port 628 (protocol 'tcp') (service 'qmqp') is open
>>> > port 953 (protocol 'tcp') (service 'rndc') is open
>>> > port 993 (protocol 'tcp') (service 'imaps') is open
>>> > port 995 (protocol 'tcp') (service 'pop3s') is open
>>> > port 1723 (protocol 'tcp') (service 'pptp') is open
>>> > port 2000 (protocol 'tcp') (service 'cisco-sccp') is open
>>> > port 2049 (protocol 'tcp') (service 'nfs') is open
>>> > port 3128 (protocol 'tcp') (service 'squid-http') is open
>>> > port 4369 (protocol 'tcp') (service 'epmd') is open
>>> > port 5038 (protocol 'tcp') (service 'unknown') is open
>>> > port 5222 (protocol 'tcp') (service 'xmpp-client') is open
>>> > port 5223 (protocol 'tcp') (service 'hpvirtgrp') is open
>>> > port 5269 (protocol 'tcp') (service 'xmpp-server') is open
>>> > port 5280 (protocol 'tcp') (service 'xmpp-bosh') is open
>>> > port 5281 (protocol 'tcp') (service 'unknown') is open
>>> > port 5432 (protocol 'tcp') (service 'postgresql') is open
>>> > port 5433 (protocol 'tcp') (service 'pyrrho') is open
>>> > port 8010 (protocol 'tcp') (service 'xmpp') is open
>>> > port 8731 (protocol 'tcp') (service 'unknown') is open
>>> > port 8888 (protocol 'tcp') (service 'sun-answerbook') is open
>>> > port 9571 (protocol 'tcp') (service 'unknown') is open
>>> > port 9572 (protocol 'tcp') (service 'unknown') is open
>>> > port 34103 (protocol 'tcp') (service 'unknown') is open
>>> > port 38602 (protocol 'tcp') (service 'unknown') is open
>>> > port 39734 (protocol 'tcp') (service 'unknown') is open
>>> > port 40032 (protocol 'tcp') (service 'unknown') is open
>>> > port 46700 (protocol 'tcp') (service 'unknown') is open
>>> > port 55266 (protocol 'tcp') (service 'unknown') is open
>>>
>> Description:

VOIP

> Parameter: VOIP Status
>> Value:
>>> OK
>> Description:
>>> The asterisk server is running.
> Parameter: Ports state
>> Value:
>>> The SIP port is closed
>>> The SIP port is closed
>>>
>> Description:

> Parameter: Registered Phones
  >> Value:
    >>> Number of registered phones: 0
    >>> Number of online phones: 0
  >> Description:

> Parameter: Objects
  >> Value:
    >>> Number of objects: 2
  >> Description:
    >>> Object Name: T38modem2
    >>> Object Type: peer
    >>> Object flag: 0
    >>> Object RefCount: 2
    >>>
    >>> Object Name: kamailio
    >>> Object Type: peer
    >>> Object flag: 0
    >>> Object RefCount: 2
    >>>
    >>>
> Parameter: Number of active calls
  >> Value:
    >>> 0
  >> Description:

> Parameter: Trunk
  >> Value:
    >>> Connectable Trunks: 0
    >>> Unconnectable Trunks: 0
    >>> Unknown Trunks: 0
    >>>
  >> Description:
    >>> Connectable Trunks:
    >>> None
    >>> Unconnectable Trunks:
    >>> None
    >>> Unknown Trunks:
    >>> None
    >>>
> Parameter: G729 License
  >> Value:
    >>> Licensed channels: 0
  >> Description:
    >>> Encoders in use: 0
    >>> Decoders in use: 0
> Parameter: Codecs
  >> Value:
    >>> Codecs used now: No codecs in use.
  >> Description:

> Parameter: Agents
  >> Value:
    >>> No Agents are configured in agents.conf
  >> Description:

> Parameter: Agents Online
  >> Value:
    >>> No Agents are configured in agents.conf
  >> Description:

> Parameter: Do not disturb
  >> Value:
    >>> 0 results found.
  >> Description:

> Parameter: Telephony cards
  >> Value:
    >>> PRI span 1/0: In Alarm, Down, Active
    >>> PRI span 2/0: In Alarm, Down, Active
    >>>
  >> Description:
    >>> ---------------------------
    >>> Port 1:
    >>> Primary D-channel: 16
    >>> Status: In Alarm , Down , Active
    >>> Switchtype: EuroISDN
    >>> Type: CPE
    >>> Remote type: Network
    >>> Overlap Dial: 1
    >>> Logical Channel Mapping: 0
    >>> Timer and counter settings:
    >>> N200: 3
    >>> N202: 3
    >>> K: 7
    >>> T200: 1000
    >>> T201: 1000

>>> T202: 10000
>>> T203: 10000
>>> T303: 4000
>>> T305: 30000
>>> T308: 4000
>>> T309: 6000
>>> T312: 6000
>>> T313: 4000
>>> T316: -1
>>> N316: 2
>>> T-HOLD: 4000
>>> T-RETRIEVE: 4000
>>> T-RESPONSE: 4000
>>> T-STATUS: 4000
>>> T-ACTIVATE: 10000
>>> T-DEACTIVATE: 4000
>>> T-INTERROGATE: 4000
>>> T-RETENTION: 30000
>>> T-CCBS1: 4000
>>> T-CCBS2: 2700000
>>> T-CCBS3: 20000
>>> T-CCBS4: 5000
>>> T-CCBS5: 3600000
>>> T-CCBS6: 3600000
>>> T-CCNR2: 10800000
>>> T-CCNR5: 11700000
>>> T-CCNR6: 11700000
>>> Q931 RX: 0
>>> Q931 TX: 0
>>> Q921 RX: 12983
>>> Q921 TX: 20015
>>> Q921 Outstanding: 0 (TEI=0)
>>> Total active-calls:0 global:0
>>> CC records:
>>> Overlap Recv: Yes
>>> ----------
>>>
>>> ----------
>>> Port 2:
>>> Primary D-channel: 47
>>> Status: In Alarm , Down , Active
>>> Switchtype: EuroISDN
>>> Type: Network
>>> Remote type: CPE
>>> Overlap Dial: 1
>>> Logical Channel Mapping: 0
>>> Timer and counter settings:
>>> N200: 3
>>> N202: 3
>>> K: 7
>>> T200: 1000
>>> T201: 1000
>>> T202: 10000
>>> T203: 10000
>>> T303: 4000
>>> T305: 30000
>>> T308: 4000
>>> T309: 6000
>>> T312: 6000
>>> T313: 4000
>>> T316: -1
>>> N316: 2
>>> T-HOLD: 4000
>>> T-RETRIEVE: 4000
>>> T-RESPONSE: 4000
>>> T-STATUS: 4000
>>> T-ACTIVATE: 10000
>>> T-DEACTIVATE: 4000
>>> T-INTERROGATE: 4000
>>> T-RETENTION: 30000
>>> T-CCBS1: 4000
>>> T-CCBS2: 2700000
>>> T-CCBS3: 20000
>>> T-CCBS4: 5000
>>> T-CCBS5: 3600000
>>> T-CCBS6: 3600000
>>> T-CCNR2: 10800000
>>> T-CCNR5: 11700000
>>> T-CCNR6: 11700000
>>> Q931 RX: 0
>>> Q931 TX: 0
>>> Q921 RX: 12988
>>> Q921 TX: 20019
>>> Q921 Outstanding: 0 (TEI=0)
>>> Total active-calls:0 global:0
>>> CC records:
>>> Overlap Recv: Yes
>>> ----------

>>>
>>>

E-mail

> Parameter: QMAIL status
>> Value:
>>> All qmail processes are running.
>> Description:

> Parameter: Queue
>> Value:
>>> Number of queued messages: 1
>>> Number of messages not pre processed: 1
>> Description:

> Parameter: Smtp Routes
>> Value:
>>> No SMTP routes configured
>> Description:

> Parameter: Internet Connection
>> Value:
>>> OK
>> Description:
>>> Tested with ip 8.8.8.8
> Parameter: Send Limit
>> Value:
>>> 40000000 Bytes
>> Description:

> Parameter: Forward Limit
>> Value:
>>> 20000000 Bytes
>> Description:

> Parameter: LDAP users
>> Value:
>>> User teste:
>>> User teste not found in Ldap
>> Description:

> Parameter: Messages status
>> Value:
>>> ---
>>> Message number 919566
>>> From: <administrator@domain.com>
>>> To: remote config.backup@ipbrick.com
>>> Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402589588_qp_13134_by_proxy2.iportalmais.pt/
>>> ---
>>>
>>> ---
>>> Message number
>>> From:
>>> To:
>>> Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402589588_qp_13135_by_proxy2.iportalmais.pt/
>>> ---
>>>
>>> ---
>>> Message number 919552
>>> From: <administrator@domain.com>
>>> To: remote config.backup@ipbrick.com
>>> Status: success: 195.23.114.203_accepted_message./Remote_host_said:_250_ok_1402596359_qp_27921_by_proxy2.iportalmais.pt/
>>> ---
>>>
>>>
>> Description:

> Parameter: POP status
>> Value:
>>> OK
>> Description:
>>> Pop service is running.
> Parameter: POPs status
>> Value:
>>> OK
>> Description:
>>> Pops service is running.
> Parameter: POP connectivity
>> Value:
>>> OK
>> Description:
>>> Pop connection ok.
> Parameter: IMAP status
>> Value:
>>> OK
>> Description:
>>> Imap service is running.

> Parameter: IMAPs status
>> Value:
>>> OK
>> Description:
>>> Imaps service is running.
> Parameter: IMAP connectivity
>> Value:
>>> OK
>> Description:
>>> Imap connection ok.

Instant Message

> Parameter: IM Status
>> Value:
>>> OK
>> Description:
>>> Ejabberd service is running.
> Parameter: IM connectivity
>> Value:
>>> OK
>> Description:
>>> Ejabberd telnet connection ok.

Web Server

> Parameter: Apache status
>> Value:
>>> OK
>> Description:
>>> Apache service is running.
> Parameter: Apache statistics
>> Value:
>>> Server Version: Apache/2.2.22 (Debian) proxy_html/3.0.1 mod_ssl/2.2.22
>>> Server Built: Feb 1 2014 21:26:04
>>> Current Time: Friday, 13-Jun-2014 16:15:18 BST
>>> Restart Time: Thursday, 12-Jun-2014 17:04:05 BST
>>> Parent Server Generation: 1
>>> Server uptime: 23 hours 11 minutes 12 seconds
>>> Total accesses: 11990 - Total Traffic: 8.9 MB
>>> CPU Usage: u2103.77 s226.57 cu3.19 cs0 - 2.8% CPU load
>>> cache type: SHMCB, shared memory: 512000 bytes, current sessions: 0
>>> subcaches: 32, indexes per subcache: 133
>>> index usage: 0%, cache usage: 0%
>>> total sessions stored since starting: 0
>>> total sessions expired since starting: 0
>>> total (pre-expiry) sessions scrolled out of the cache: 0
>>> total retrieves since starting: 0 hit, 68 miss
>>> total removes since starting: 0 hit, 0 miss
>> Description:

> Parameter: Virtual hosts
>> Value:
>>> Port: 80
>>> Name: ipbrick.domain.com
>>> File: (/etc/apache2/sites-enabled/200-1-ipbrick.domain.com:42)
>>>
>>> Port: 80
>>> Name: groupware.domain.com
>>> File: (/etc/apache2/sites-enabled/200-100-groupware.domain.com:37)
>>>
>>> Port: 80
>>> Name: wpad.domain.com
>>> File: (/etc/apache2/sites-enabled/200-112-wpad.domain.com:6)
>>>
>>> Port: 80
>>> Name: ipbrick4cc.domain.com
>>> File: (/etc/apache2/sites-enabled/200-118-ipbrick4cc.domain.com:18)
>>>
>>> Port: 80
>>> Name: broker.domain.com
>>> File: (/etc/apache2/sites-enabled/200-119-broker.domain.com:6)
>>>
>>> Port: 80
>>> Name: cafe.domain.com
>>> File: (/etc/apache2/sites-enabled/200-128-cafe.domain.com:37)
>>>
>>> Port: 80
>>> Name: ucoip.domain.com
>>> File: (/etc/apache2/sites-enabled/200-200-ucoip.domain.com:6)
>>>
>>> Port: 80
>>> Name: contacts.domain.com
>>> File: (/etc/apache2/sites-enabled/200-64-contacts.domain.com:6)
>>>
>>> Port: 80
>>> Name: pgsqladmin.domain.com
>>> File: (/etc/apache2/sites-enabled/200-66-pgsqladmin.domain.com:6)

>>>
>>> Port: 80
>>> Name: callmanager.domain.com
>>> File: (/etc/apache2/sites-enabled/200-71-callmanager.domain.com:6)
>>>
>>> Port: 80
>>> Name: jwchat.domain.com
>>> File: (/etc/apache2/sites-enabled/200-72-jwchat.domain.com:6)
>>>
>>> Port: 80
>>> Name: webphone.domain.com
>>> File: (/etc/apache2/sites-enabled/200-73-webphone.domain.com:6)
>>>
>>> Port: 80
>>> Name: callstatistics.domain.com
>>> File: (/etc/apache2/sites-enabled/200-74-callstatistics.domain.com:28)
>>>
>>> Port: 443
>>> Name: ipbrick.domain.com
>>> File: (/etc/apache2/sites-enabled/200-1-ipbrick.domain.com:6)
>>>
>>> Port: 443
>>> Name: groupware.domain.com
>>> File: (/etc/apache2/sites-enabled/200-100-groupware.domain.com:6)
>>>
>>> Port: 443
>>> Name: ipbrick4cc.domain.com
>>> File: (/etc/apache2/sites-enabled/200-118-ipbrick4cc.domain.com:6)
>>>
>>> Port: 443
>>> Name: cafe.domain.com
>>> File: (/etc/apache2/sites-enabled/200-128-cafe.domain.com:6)
>>>
>>> Port: 443
>>> Name: callstatistics.domain.com
>>> File: (/etc/apache2/sites-enabled/200-74-callstatistics.domain.com:6)
>>>
>>>
>> Description:

> Parameter: Configuration files
>> Value:
>>> Apache configuration files without errors.
>> Description:
>>>

Proxy

> Parameter: Squid status
>> Value:
>>> OK
>> Description:
>>> Proxy service is running.
> Parameter: Squid cache
>> Value:
>>> Squid cache size is 0 Bytes - 0% of the limit
>> Description:
>>> Used cache: 0 Bytes
>>> Cache Limit: 335544320 Bytes
>>> Cache used percentage: 0 %
> Parameter: Squid check page
>> Value:
>>> OK
>> Description:
>>> The page http://www.google.pt is available with proxy localhost:3128
> Parameter: Dansguardian status
>> Value:
>>> CRITICAL
>> Description:
>>> Dansguardian service is not running.
> Parameter: Dansguardian blocked urls
>> Value:
>>>
>> Description:

> Parameter: Configuration files
>> Value:
>>> No errors in squid configuration file.
>> Description:

vpn

> Parameter: Ipsec Verify
>> Value:
>>> Checking your system to see if IPsec got installed and started correctly:
>>> Version check and ipsec on-path OK
>>> Linux Openswan U2.6.37-g955aaafb-dirty/K3.2.0-4-amd64 (netkey)

>>> Checking for IPsec support in kernel OK
>>> SAref kernel support N/A
>>> NETKEY: Testing XFRM related proc values FAILED
>>> Please disable /proc/sys/net/ipv4/conf/*/send_redirects
>>> or NETKEY will cause the sending of bogus ICMP redirects!
>>> FAILED
>>> Please disable /proc/sys/net/ipv4/conf/*/accept_redirects
>>> or NETKEY will accept bogus ICMP redirects!
>>> OK
>>> Checking that pluto is running FAILED
>>> whack: Pluto is not running (no "/var/run/pluto/pluto.ctl")
>>> Two or more interfaces found, checking IP forwarding FAILED
>>> whack: Pluto is not running (no "/var/run/pluto/pluto.ctl")
>>> Checking NAT and MASQUERADEing OK
>>> Checking for 'ip' command OK
>>> Checking /bin/sh is not /bin/dash OK
>>> Checking for 'iptables' command OK
>>> Opportunistic Encryption Support DISABLED
>>>
>> Description:

> Parameter: Ipsec Status
>> Value:
>>> whack: Pluto is not running (no "/var/run/pluto/pluto.ctl")
>>>
>> Description:

Referências

[1] Ipbrick International. IPBRICK.IC. visitada em 12/12/13. URL: http://www.ipbrick.pt/index.php?oid=323.

[2] How Does DNS Work? visitada em 12/12/13. URL: http://oreilly.com/catalog/dns3/chapter/ch02.html.

[3] DHCP (Dynamic Host Configuration Protocol). visitada em 12/12/13. URL: http://study-ccna.com/dhcp-dns.

[4] Ipbrick International. IPBRICK SA. visitada em 12/12/13. URL: http://www.ipbrick.com.

[5] Asterisk - Critical Architectural Concepts. visitada em 2/2/14. URL: http://aosabook.org/en/asterisk.html.

[6] Correio eletrónico (E-mail). visitada em 5/2/14. URL: http://www.inescporto.pt/~jneves/feup/2013-2014/scom/1.pdf.

[7] Proxy server. visitada em 5/2/14. URL: http://www.icc.sa/icc2012/icc_site/Products.aspx?d=11.

[8] Nagios Community. Plugins. visitada em 29/12/13. URL: http://nagios.sourceforge.net/docs/3_0/plugins.html.

[9] Nagios Community. Configuration Overview. visitada em 29/12/13. URL: http://nagios.sourceforge.net/docs/3_0/config.html.

[10] Nagios Monitoring Agents. NRPE. visitada em 29/12/13. URL: http://exchange.nagios.org/directory/Addons/Monitoring-Agents/NRPE--2D-Nagios-Remote-Plugin-Executor/details.

[11] Nagios Community. Monitoring Windows Machines . visitada em 29/12/13. URL: http://nagios.sourceforge.net/docs/3_0/monitoring-windows.html.

[12] João Neves. The Simple Network Management Protocol, version 1. visitada em 21/04/14. URL: http://www.inescporto.pt/~jneves/feup/2011-2012/pgre/snmp.pdf.

[13] Nagios Community. Plugins API. visitada em 29/12/13. URL: http://nagios.sourceforge.net/docs/3_0/pluginapi.html.

[14] Linux Debian. Debian. visitada em 05/02/14. URL: https://www.debian.org/.

[15] Ipbrick International. Reference Guide, 2013.

[16] Cenários IPBrick. IPBrick. visitada em 12/12/13. URL: http://www.ipbrick.pt/index.php?oid=1155.

[17] IETF. Lightweight directory access protocol (ldap):technical specification road map, June 2006. RFC 4510.

[18] IETF. Simple mail transfer protocol, April 2001. RFC 2821.

[19] IETF. Hypertext transfer protocol – http/1.1, June 1999. RFC 2616.

[20] Apache. Apache. visitada em 12/12/13. URL: http://www.apache.org/.

[21] PostgreSQL. PostgreSQL. visitada em 12/12/13. URL: http://www.postgresql.org/.

[22] OpenLDAP. OpenLDAP. visitada em 12/12/13. URL: http://www.openldap.org/.

[23] The SQL Language. visitada em 12/12/13. URL: http://www.postgresql.org/docs/8.1/static/sql.html.

[24] MySQL. MySQL. visitada em 12/12/13. URL: http://www.mysql.com/.

[25] PostgreSQL history. visitada em 12/12/13. URL: http://www.postgresql.org/about/history/.

[26] IETF. Domain names - concepts and facilities, November 1987. RFC 1034.

[27] DNS, BIND, DHCP, LDAP and Directory Services. visitada em 2/2/14. URL: http://www.bind9.net/.

[28] IETF. Dynamic host configuration protocol, March 1997. RFC 2131.

[29] The netfilter.org project - IPTables. visitada em 2/2/14. URL: http://www.netfilter.org/.

[30] Asterisk. visitada em 2/2/14. URL: http://www.asterisk.org/.

[31] Asterisk - Contexts, Extensions, and Priorities. visitada em 2/2/14. URL: https://wiki.asterisk.org/wiki/display/AST/Contexts,+Extensions,+and+Priorities.

[32] The Asterisk Dialplan. visitada em 2/2/14. URL: https://wiki.asterisk.org/wiki/display/AST/The+Asterisk+Dialplan.

[33] Kamailio SIP Server. visitada em 2/2/14. URL: http://www.kamailio.org/.

[34] Mozilla Thunderbird. visitada em 5/2/14. URL: http://www.mozilla.org/pt-PT/thunderbird/.

[35] Microsoft Outlook. visitada em 5/2/14. URL: http://office.microsoft.com/pt-pt/microsoft-outlook-software-de-e-mail-e-calendario-FX010048775.aspx.

[36] Qmail. visitada em 5/2/14. URL: http://cr.yp.to/qmail.html.

[37] Courier Mail Server . visitada em 5/2/14. URL: http://www.courier-mta.org/.

[38] IETF. The secure sockets layer (ssl) protocol version 3.0, August 2011. RFC 6101.

[39] Ejabberd . visitada em 5/2/14. URL: http://www.ejabberd.im/.

[40] HylaFAX The world's most advanced open source fax server . visitada em 5/2/14. URL: http://www.hylafax.org/.

[41] Manual online IPBrick. visitada em 5/2/14. URL: http://www.ipbrick.com/pt/documentacao/manual_referenciaV4.0/node33.html.

[42] Squid: Optimising Web Delivery. visitada em 5/2/14. URL: http://www.squid-cache.org/.

[43] Zabbix SIA. Zabbix. visitada em 12/12/13. URL: http://www.zabbix.com.

[44] Zenoss, Inc. . Zenoss. visitada em 12/12/13. URL: http://www.zenoss.com.

[45] Nagios Official Site. Community. visitada em 13/12/13. URL: http://www.nagios.com/about/community.

[46] Nagios. Nagios - The Industry Standard in IT Infrastructure Monitoring. visitada em 13/12/13. URL: http://www.nagios.org/.

[47] Nagios Official Site. History. visitada em 13/12/13. URL: http://www.nagios.org/about/history.

[48] Perl Official Site. PERL language. visitada em 13/12/13. URL: http://www.perl.org/.

[49] PHP. Official documentation. visitada em 14/12/13. URL: http://www.php.net/manual/en/.

[50] C Programming and C++ Programming. visitada em 5/2/14. URL: http://www.cprogramming.com/.

[51] Nagios Monitoring Agents. NSClient++. visitada em 29/12/13. URL: http://exchange.nagios.org/directory/Addons/Monitoring-Agents/NSClient/details.

[52] IETF. Management information base (mib) for the simple network management protocol (snmp), December 2002. RFC 3418.

[53] Free Software Foundation, Inc. . GNU GPL FAQ. visitada em 14/03/14. URL: https://www.gnu.org/licenses/gpl-faq.html.

[54] Zend Technologies Ltd. Zend Guard Encode Your PHP Applications. visitada em 14/03/14. URL: http://www.zend.com/en/products/guard/.

[55] Linux / Unix Command: netstat. visitada em 5/2/14. URL: http://linux.about.com/od/commands/l/blcmdl8_netstat.htm.